

APPROXIMATE ANALYTICAL EVALUATION OF
EXTENDED-KALMAN FILTERS

Hasan Öner Tasdelen

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93940

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

APPROXIMATE ANALYTICAL EVALUATION
OF EXTENDED-KALMAN FILTERS

by

Hasan Öner Taşdelen

December 1975

Thesis Advisor:

D. E. Kirk

Approved for public release; distribution unlimited.

T170472

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Approximate Analytical Evaluation of Extended-Kalman Filters		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; December 1975
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Hasan Oner Tasdelen		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE December 1975
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) **Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Analytic equations, extended-Kalman filter, performance evaluation, Monte-Carlo simulation, conditional expectation.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Analytical equations derived for evaluating linear estimators are applied to extended-Kalman filters for approximate perform- ance evaluation. Two cases were considered, a single known target trajectory and multiple target trajectories with given probabilities of occurrence. For the multiple-trajectory case, equations are derived for the mean and covariance of estimation error in terms of the conditional expectations. Two examples		

are presented to compare the use of the analytical equations with Monte-Carlo simulation.

Approximate Analytical Evaluation
of
Extended-Kalman Filters

by

Hasan Öner Taşdelen
Lieutenant, Turkish Navy

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1975

ABSTRACT

Analytical equations derived for evaluating linear estimators are applied to extended-Kalman filters for approximate performance evaluation. Two cases were considered, a single known target trajectory and multiple target trajectories with given probabilities of occurrence. For the multiple-trajectory case, equations are derived for the mean and covariance of estimation error in terms of the conditional expectations. Two examples are presented to compare the use of the analytical equations with Monte-Carlo simulation.

TABLE OF CONTENTS

I.	INTRODUCTION -----	10
II.	ANALYTICAL EQUATIONS -----	12
	A. ASSUMPTIONS -----	12
	B. DERIVATION OF ANALYTICAL EQUATIONS -----	14
III.	APPLICATION OF THE ANALYTICAL EQUATIONS TO THE CASE OF MORE THAN ONE TRACK -----	26
	A. CONDITIONAL EXPECTATION -----	27
	B. A LINEAR PROBLEM -----	29
IV.	ESTIMATORS FOR NONLINEAR MODELS -----	40
	A. DISCRETIZATION -----	40
	B. LINEARIZATION -----	41
	C. THE EXTENDED-KALMAN FILTER -----	43
V.	APPLICATION OF THE ANALYTICAL EQUATIONS TO EXTENDED-KALMAN FILTERS: A RE-ENTRY PROBLEM ---	46
	A. THE RE-ENTRY PROBLEM -----	46
	B. APPLICATION OF THE ANALYTICAL APPROACH TO THE MULTIPLE TRACK NONLINEAR CASE -----	79
	C. CONDITIONAL COVARIANCE -----	80
VI.	A SUBMARINE TRACKING PROBLEM -----	92
VII.	CONCLUSIONS -----	112
	APPENDIX A: Computer Programs -----	114
	A. Program "Eval" -----	114
	B. Monte-Carlo Simulation -----	117
	COMPUTER PROGRAM -----	118
	BIBLIOGRAPHY -----	173
	INITIAL DISTRIBUTION LIST -----	174

LIST OF FIGURES

3.1.	Time history of mean position -----	34
3.2.	Time history of mean velocity -----	35
3.3.	Position estimation error history -----	36
3.4.	Velocity estimation error history -----	37
3.5.	Variance of position estimation error vs. time -----	38
3.6.	Variance of velocity estimation error vs. time -----	39
5.1.	Geometry of Re-entry problem -----	47
5.2.	Altitude history of re-entry vehicle -----	54
5.3.	Velocity history of re-entry vehicle -----	55
5.4.	Time history of ballistic coefficient -----	56
5.5.	Time history of altitude estimation error ---	57
5.6.	Time history of the velocity estimation error -----	58
5.7.	Time history of the mean ballistic parameter estimation error -----	59
5.8.	Variance of altitude estimation error vs. time -----	60
5.9.	Variance of velocity estimation error vs. time -----	61
5.10.	Variance of ballistic parameter estimation error vs. time -----	62
5.11.	Time history of the mean altitude estimation error for 50 Monte-Carlo runs -----	66
5.12.	Time history of the mean velocity estimation error for 50 Monte-Carlo runs -----	67
5.13.	Time history of the mean ballistic parameter estimation error for 50 Monte-Carlo runs ----	68

5.14.	Variance of the altitude estimation error vs. time for 50 Monte-Carlo runs -----	69
5.15.	Variance of the velocity estimation error vs. time for 50 Monte-Carlo runs -----	70
5.16.	Variance of the ballistic parameter estimation error vs. time for 50 Monte-Carlo runs -----	71
5.17.	Time history of altitude estimation error for $\underline{X}(0) = \underline{b}$ -----	73
5.18.	Time history of mean velocity estimation error for $\underline{X}(0) = \underline{b}$ -----	74
5.19.	Time history of the mean ballistic parameter estimation error for $\underline{X}(0) = \underline{b}$ -----	75
5.20.	Variance of altitude estimation error vs. time for $\underline{X}(0) = \underline{b}$ -----	76
5.21.	Variance of velocity estimation error vs. time for $\underline{X}(0) = \underline{b}$ -----	77
5.22.	Variance of ballistic parameter estimation error vs. time for $\underline{X}(0) = \underline{b}$ -----	78
5.23.	Time history of the mean altitude estimation error for two tracks -----	86
5.24.	Time history of the mean velocity estimation error for two tracks -----	87
5.25.	Time history of the ballistic parameter estimation error for two tracks -----	88
5.26.	Time history of the variance of the altitude estimation error for two tracks -----	89
5.27.	Time history of the variance of the velocity estimation error for two tracks -----	90
5.28.	Time history of the variance of the ballistic parameter estimation error for two tracks ---	91
6.1.	$R_{cpa} - X_{cpa}$ filter geometry -----	93
6.2.	Time history of the mean R_{cpa} estimation error -----	102

6.3.	Time history of the mean X_{cpa} estimation error -----	103
6.4.	Time history of the mean V_s estimation error-----	104
6.5.	Time history of the mean θ_s estimation error -----	105
6.6.	Time history of the mean F_o estimation error -----	106
6.7.	Variance of the mean R_{cpa} estimation error vs. time -----	107
6.8.	Variance of the mean X_{cpa} estimation error vs. time -----	108
6.9.	Variance of the mean V_s estimation error vs. time -----	109
6.10.	Variance of the mean θ_s estimation error vs. time -----	110
6.11.	Variance of the mean F_o estimation error vs. time -----	111

ACKNOWLEDGEMENT

The author is deeply indebted to the United States Navy and the Turkish Navy for providing the opportunity to pursue this work.

I wish to thank my thesis advisor Professor D. E. Kirk and all my professors, for their guidance and counsel.

I. INTRODUCTION

In state estimation problems one of the tasks is the simulation of a filter or estimator after designing it. The purpose of the simulation is to determine the performance of the filter under actual operating conditions and to investigate sensitivity to inaccuracies or approximations present in the design assumptions.

Monte-Carlo simulation is a common simulation algorithm which is currently used. An important drawback of the Monte-Carlo approach is the large amounts of computer time generally required to achieve a reasonable degree of accuracy. To reduce the computation time requirements, fewer Monte-Carlo runs maybe used with an attendant loss of accuracy. Because of these limitations of the Monte-Carlo approach, an alternative method, cosisting of a set of analytical equations, has been derived and a computer algorithm has been established for the evaluation of estimation and prediction error statistics for linear filters. It is possible to characterize the propagation of the means and covariances of estimation error of a filter by difference equations. These difference equations can easily be solved using relatively small amounts of computer time. The only disadvantage of the analytical equation approach is that it only applies to linear filters with precomputed gain schedules.

In practice, one may be confronted by nonlinear problems, such as space vehicle re-entry or orbit determination problems, or fire-control estimation problems. A common approach for nonlinear problems is to use an extended Kalman filter and to evaluate its performance by Monte-Carlo simulation.

The main objective of this research is to investigate the possibility of approximate evaluation of the performance of extended Kalman filters by applying the analytical equation approach. In Chapter II of this thesis the analytical equations are derived for linear time-invariant estimators. In Chapter III, the application of the analytical equations to problems with multiple tracks occurring with given probabilities is investigated and an example is presented. In Chapter IV, a common filtering approach for nonlinear systems is summarized. In Chapter V, the application of the analytical equations to extended Kalman filters is discussed for both the single and multiple track cases by using a nonlinear example. Chapter VI contains simulation results from another example. The simulations were performed using both the analytical equations and the Monte-Carlo algorithm for comparison.

The computer programs that were used are given in Appendix A.

II. ANALYTICAL EQUATIONS

A. ASSUMPTIONS

In the development of analytical equations the following assumptions have been made [1] :

1. The true state trajectory $\underline{X}(k)$ for $k=0,1,2,\dots$ is known.

2. The measurement equation is

$$\underline{Z}(k) = \underline{h}(\underline{X}(k)) + \underline{V}(k) \quad (2.1)$$

where

$\underline{Z}(k)$ is the q -dimensional measurement vector at time $t=kT$.

\underline{h} is a function (which may be nonlinear) of $\underline{X}(k)$.

$\underline{V}(k)$ is the measurement noise vector with the assumptions

$$(a). \quad E[\underline{V}(k)] = \underline{0} \text{ for all } k \geq 0.$$

(The expected value of the measurement noise is zero for all time)

$$(b). \quad E[\underline{V}(k)\underline{V}(j)^T] = \begin{cases} \underline{R}(k) & \text{for } k=j \\ \underline{0} & \text{for } k \neq j \end{cases} \text{ for all } k, j \geq 0$$

(the measurement noise is uncorrelated).

(c). Since the target trajectory is known

$$E[\underline{X}(j)\underline{V}(k)^T] = \underline{X}(j).E[\underline{V}(k)^T] = \underline{0} \text{ for all } k, j \geq 0$$

3. The estimator is linear and described by the equation

$$\hat{\underline{X}}(k/k) = \hat{\underline{X}}(k/k-1) + \underline{Q}(k) [\underline{Z}(k) - \underline{C}\hat{\underline{X}}(k/k-1)] \quad (2.2)$$

The prediction equation is

$$\hat{\underline{X}}(k/k-1) = \phi \hat{\underline{X}}(k-1/k-1) + \Delta \underline{U}(k-1) \quad (2.3)$$

where

$\hat{\underline{X}}(k/k)$ is the n -dimensional estimate of $\underline{X}(k)$

given measurements

$\underline{Z}(0), \underline{Z}(1), \dots, \underline{Z}(k)$.

$\hat{\underline{X}}(k/k-1)$ is the predicted value of $\underline{X}(k)$ given

measurements $\underline{Z}(0), \underline{Z}(1), \dots, \underline{Z}(k-1)$.

ϕ and Δ are $n \times n$ and $q \times n$ matrices, respectively, and they are assumed to be known.

$\underline{U}(k-1)$ is the m -dimensional deterministic forcing vector at time $t = (k-1)T$.

Δ is an $n \times m$ known matrix.

$\underline{G}(k)$ is the $n \times m$ gain matrix.

The gains may be found by any method. For example, optimal estimation gains which minimize the sum of the variance of estimation error can be obtained by using the Kalman equations:

$$\underline{G}(k) = \underline{P}(k/k-1) \underline{C}^T \left[\underline{C} \underline{P}(k/k-1) \underline{C}^T + \underline{R}(k) \right]^{-1} \quad (2.4)$$

$$\underline{P}(k/k) = \left[\underline{I} - \underline{G}(k) \underline{C} \right] \underline{P}(k/k-1) \quad (2.5)$$

$$\underline{P}(k/k-1) = \phi \underline{P}(k-1/k-1) \phi^T + \underline{Q}(k-1) \quad (2.6)$$

with the initial conditions

$$\underline{P}(0/-1) = \underline{P}_0 = E \left\{ \left[\underline{X}(0) - \underline{X}_0 \right] \cdot \left[\underline{X}(0) - \underline{X}_0 \right]^T \right\}$$

$$\hat{\underline{X}}(0/-1) = \underline{\bar{X}}_0$$

(In this research gains determined by using the

Kalman equations have been used for simulations.)

Where

$$\underline{P}(k/k) = E \left\{ \left[\hat{\underline{X}}(k/k) - \underline{X}(k) \right] \cdot \left[\hat{\underline{X}}(k/k) - \underline{X}(k) \right]^T \right\} \quad (2.7)$$

which is the covariance matrix of theoretical

estimation error, and

$$\underline{P}(k/k-1) = E \left\{ \left[\hat{\underline{X}}(k/k-1) - \underline{X}(k) \right] \cdot \left[\hat{\underline{X}}(k/k-1) - \underline{X}(k) \right]^T \right\} \quad (2.8)$$

$\underline{Q}(k)$ is the covariance matrix of the random forcing input $\underline{W}(k)$. i.e.

$$E \left[\underline{W}(k) \underline{W}(j)^T \right] = \begin{cases} \underline{Q}(k) & \text{for } k=j \\ \underline{0} & \text{for } k \neq j \end{cases} \quad (2.9)$$

If the estimation equation (2.2) is initialized with the value

$$\hat{\underline{X}}(0/-1) = \underline{X}_0 = E \underline{X}(0) \quad (2.10)$$

it can be shown that the optimal estimate $\hat{\underline{X}}(k/k)$ is unbiased, i.e.

$$E \left[\hat{\underline{X}}(k/k) - \underline{X}(k) \right] = \underline{0} \quad (2.11)$$

With the assumptions given above one can derive

difference equations for the mean of estimation error, the covariance of estimation error, the mean of N-step prediction error and the covariance of N-step prediction error. These equations are derived for the time-invariant estimator given by equations (2.2) and (2.3) in the following section.

B. DERIVATION OF ANALYTICAL EQUATIONS

1. Mean and covariance of estimation error

The true state of the target at time k is $\underline{X}(k)$; thus, using Equations (2.2) and (2.3) the estimation error $\tilde{\underline{X}}(k)$ at time k is

$$\begin{aligned}
\tilde{\underline{X}}(k) &\triangleq \hat{\underline{X}}(k/k) - \underline{X}(k) \\
&= \varnothing \hat{\underline{X}}(k-1/k-1) + \underline{\Delta} \underline{U}(k-1) + \underline{G}(k) \left[\underline{Z}(k) \right. \\
&\quad \left. - \underline{C} \varnothing \underline{X}(k-1/k-1) - \underline{C} \underline{\Delta} \underline{U}(k-1) \right] - \underline{X}(k) \quad (2.12)
\end{aligned}$$

Substituting Equation (2.1) into (2.12) gives

$$\begin{aligned}
\tilde{\underline{X}}(k) &= \left[\underline{I} - \underline{G}(k) \underline{C} \right] \varnothing \hat{\underline{X}}(k-1/k-1) + \underline{G}(k) \underline{V}(k) \\
&\quad + \underline{G}(k) \underline{h}(\underline{X}(k)) - \underline{X}(k) \\
&\quad + \left[\underline{I} - \underline{G}(k) \underline{C} \right] \underline{\Delta} \underline{U}(k-1) \quad (2.13)
\end{aligned}$$

By defining the deterministic matrices $\underline{S}(k)$ and $\underline{D}(k)$ as

$$\underline{S}(k) = \left[\underline{I} - \underline{G}(k) \underline{C} \right] \varnothing \quad (2.14)$$

$$\begin{aligned}
\underline{D}(k) &= \left[\underline{I} - \underline{G}(k) \underline{C} \right] \underline{\Delta} \underline{U}(k-1) + \underline{G}(k) \underline{h}(\underline{X}(k)) \\
&\quad - \underline{X}(k) \quad (2.15)
\end{aligned}$$

Equation (2.13) becomes

$$\tilde{\underline{X}}(k) = \underline{S}(k) \hat{\underline{X}}(k-1/k-1) + \underline{D}(k) + \underline{G}(k) \underline{V}(k) \quad (2.16)$$

The mean of estimation error is defined as

$$E[\tilde{\underline{X}}(k)] = E \left[\underline{S}(k) \hat{\underline{X}}(k-1/k-1) + \underline{D}(k) + \underline{G}(k) \underline{V}(k) \right]$$

The matrices $\underline{S}(k)$ and $\underline{D}(k)$ are deterministic, hence

$$E \left[\underline{G}(k) \underline{V}(k) \right] = \underline{G}(k) E \left[\underline{V}(k) \right] = \underline{0}$$

and using properties of the expectation operator gives

$$E \left[\tilde{\underline{X}}(k) \right] = \underline{S}(k) E \left[\hat{\underline{X}}(k-1/k-1) \right] + \underline{D}(k)$$

Defining

$$\tilde{\underline{\mu}}(k) = E \left[\tilde{\underline{X}}(k) \right]$$

gives

$$\tilde{\underline{\mu}}(k) = \underline{S}(k) E \left[\hat{\underline{X}}(k-1/k-1) \right] + \underline{D}(k) \quad (2.17)$$

Equation (2.16) can also be written as

$$\begin{aligned}
 \tilde{\underline{X}}(k) &= \underline{S}(k) \hat{\underline{X}}(k-1/k-1) + \underline{D}(k) + \underline{G}(k) \underline{V}(k) \\
 &\quad + \left[\underline{S}(k) \underline{X}(k-1) - \underline{S}(k) \hat{\underline{X}}(k-1/k-1) \right] \\
 &= \underline{S}(k) \left[\hat{\underline{X}}(k-1/k-1) - \underline{X}(k-1) \right] + \underline{D}(k) \\
 &\quad + \underline{G}(k) \underline{V}(k) + \underline{S}(k) \underline{X}(k-1)
 \end{aligned}$$

or

$$\begin{aligned}
 \tilde{\underline{X}}(k) &= \underline{S}(k) \tilde{\underline{X}}(k-1) + \underline{D}(k) + \underline{G}(k) \underline{V}(k) \\
 &\quad + \underline{S}(k) \underline{X}(k-1)
 \end{aligned} \tag{2.18}$$

Since $\underline{X}(k-1)$ is deterministic and

$$E \left[\tilde{\underline{X}}(k-1) \right] = \tilde{\underline{\mu}}(k-1)$$

Equation (2.17) can be written as

$$\tilde{\underline{\mu}}(k) = \underline{S}(k) \tilde{\underline{\mu}}(k-1) + \underline{D}(k) + \underline{S}(k) \underline{X}(k-1) \tag{2.19}$$

Equation (2.19) defines the mean of estimation error at time k , in terms of the mean of estimation error at time $(k-1)$.

The covariance of estimation error is defined

as

$$\begin{aligned}
 \tilde{\underline{P}}(k/k) &= E \left\{ \left[\tilde{\underline{X}}(k) - \tilde{\underline{\mu}}(k) \right] \cdot \left[\tilde{\underline{X}}(k) - \tilde{\underline{\mu}}(k) \right]^T \right\} \\
 &= E \left[\underline{Y}(k) \underline{Y}(k)^T \right]
 \end{aligned} \tag{2.20}$$

where

$$\underline{Y}(k) = \tilde{\underline{X}}(k) - \tilde{\underline{\mu}}(k) \tag{2.21}$$

One can obtain a difference equation for $\underline{Y}(k)$ by substituting (2.18) and (2.19) into (2.21), that is,

$$\begin{aligned}
 \underline{Y}(k) &= \underline{S}(k) \tilde{\underline{X}}(k-1) + \cancel{\underline{D}(k)} + \underline{G}(k) \underline{V}(k) + \cancel{\underline{S}(k) \tilde{\underline{X}}(k-1)} \\
 &\quad - \cancel{\underline{S}(k) \tilde{\underline{U}}(k-1)} - \cancel{\underline{D}(k)} - \cancel{\underline{S}(k) \tilde{\underline{X}}(k-1)} \\
 &= \underline{S}(k) \left[\tilde{\underline{X}}(k-1) - \tilde{\underline{U}}(k-1) \right] + \underline{G}(k) \underline{V}(k) \\
 &= \underline{S}(k) \underline{Y}(k-1) + \underline{G}(k) \underline{V}(k) \tag{2.22}
 \end{aligned}$$

Then

$$\begin{aligned}
 \tilde{\underline{P}}(k/k) &= E \left[\underline{Y}(k) \underline{Y}(k)^T \right] \\
 &= E \left[\underline{S}(k) \underline{Y}(k-1) \underline{Y}(k-1)^T \underline{S}(k)^T \right] \\
 &\quad + E \left[\underline{S}(k) \underline{Y}(k-1) \underline{V}(k)^T \underline{G}(k)^T \right] \\
 &\quad + E \left[\underline{G}(k) \underline{V}(k) \underline{Y}(k-1)^T \underline{S}(k)^T \right] \\
 &\quad + E \left[\underline{G}(k) \underline{V}(k) \underline{V}(k)^T \underline{G}(k)^T \right]
 \end{aligned}$$

Using properties of the expectation operator yields

$$\begin{aligned}
 \tilde{\underline{P}}(k/k) &= \underline{S}(k) E \left[\underline{Y}(k-1) \underline{Y}(k-1)^T \right] \underline{S}(k)^T \\
 &\quad + \underline{S}(k) E \left[\underline{Y}(k-1) \underline{V}(k)^T \right] \underline{G}(k)^T \\
 &\quad + \underline{G}(k) E \left[\underline{V}(k) \underline{Y}(k-1)^T \right] \underline{S}(k)^T \\
 &\quad + \underline{G}(k) \underline{R}(k) \underline{G}(k)^T \tag{2.23}
 \end{aligned}$$

To obtain expressions for $E \left[\underline{Y}(k-1) \underline{V}(k)^T \right]$ and $E \left[\underline{V}(k) \underline{Y}(k-1)^T \right]$

one can start by deriving an expression for $\underline{Y}(0)$. The

definition of $\underline{Y}(0)$ is

$$\begin{aligned}
 \underline{Y}(0) &= \tilde{\underline{X}}(0) - \tilde{\underline{U}}(0) \\
 &= \hat{\underline{X}}(0/0) - \underline{X}(0) - E \left[\tilde{\underline{X}}(0) \right]
 \end{aligned}$$

$$= \hat{\underline{X}}(0/0) - \underline{X}(0) - E \left[\hat{\underline{X}}(0/0) - \underline{X}(0) \right]$$

Since $E[\underline{X}(0)] = \underline{X}(0),$

$$\underline{Y}(0) = \hat{\underline{X}}(0/0) - E \left[\hat{\underline{X}}(0/0) \right] \quad (2.24)$$

But

$$\begin{aligned} \hat{\underline{X}}(0/0) &= \hat{\underline{X}}(0/-1) + \underline{G}(0) \left[\underline{Z}(0) - \underline{C} \hat{\underline{X}}(0/-1) \right] \\ &= \left[\underline{I} - \underline{G}(0) \underline{C} \right] \hat{\underline{X}}(0/-1) + \underline{G}(0) \underline{h}(\underline{X}(0)) \\ &\quad + \underline{G}(0) \underline{V}(0) \end{aligned} \quad (2.25)$$

Substituting (2.25) into (2.24) gives

$$\begin{aligned} \underline{Y}(0) &= \left[\underline{I} - \underline{G}(0) \underline{C} \right] \hat{\underline{X}}(0/-1) + \underline{G}(0) \underline{h}(\underline{X}(0)) \\ &\quad + \underline{G}(0) \underline{V}(0) - E \left\{ \left[\underline{I} - \underline{G}(0) \underline{C} \right] \hat{\underline{X}}(0/-1) \right. \\ &\quad \left. + \underline{G}(0) \underline{h}(\underline{X}(0)) + \underline{G}(0) \underline{V}(0) \right\} \end{aligned}$$

However,

$$E \left[\underline{G}(0) \underline{V}(0) \right] = \underline{G}(0) E \left[\underline{V}(0) \right] = 0$$

$$E \left[\underline{G}(0) \underline{h}(\underline{X}(0)) \right] = \underline{G}(0) \underline{h}(\underline{X}(0))$$

and using properties of the expectation operator gives

$$\begin{aligned} \underline{Y}(0) &= \left[\underline{I} - \underline{G}(0) \underline{C} \right] \hat{\underline{X}}(0/-1) + \cancel{\underline{G}(0) \underline{h}(\underline{X}(0))} \\ &\quad + \underline{G}(0) \underline{V}(0) - \left[\underline{I} - \underline{G}(0) \underline{C} \right] E \left[\hat{\underline{X}}(0/-1) \right] \\ &\quad - \cancel{\underline{G}(0) \underline{h}(\underline{X}(0))} \end{aligned}$$

But $\hat{\underline{X}}(0/-1)$ is a deterministic, known quantity, so

$$E \left[\hat{\underline{X}}(0/-1) \right] = \hat{\underline{X}}(0/-1) \text{ and}$$

$$\underline{Y}(0) = \underline{G}(0) \underline{V}(0) \quad (2.26)$$

Using Equation (2.22)

$$\begin{aligned}\underline{Y}(1) &= \underline{S}(1) \underline{Y}(0) + \underline{G}(1) \underline{V}(1) \\ &= \underline{S}(1) \underline{G}(0) \underline{V}(0) + \underline{G}(1) \underline{V}(1)\end{aligned}\quad (2.27)$$

and it can be shown that

$$\underline{Y}(k) = \sum_{j=0}^{k-1} \left[\prod_{i=0}^{k-j-1} \underline{S}(k-i) \right] \underline{G}(j) \underline{V}(j) + \underline{G}(k) \underline{V}(k) \quad (2.28)$$

Equation (2.28) shows that $\underline{Y}(k)$ is a linear combination of $\underline{V}(k)$ with coefficients which are known constant matrices, i.e.

$$\underline{Y}(k) = \sum_{\ell=0}^k \underline{L}_{\ell}(k) \underline{V}(\ell) \quad (2.29)$$

where

$$\begin{aligned}\underline{L}_{\ell}(k) &= \sum_{i=0}^{k-\ell-1} \underline{S}(k-i) \underline{G}(\ell) \quad \text{for } \ell=0,1,2,\dots,(k-1) \\ \underline{L}_k(k) &= \underline{G}(k)\end{aligned}\quad (2.30)$$

From Equation (2.29) it is easily seen that

$$E \left[\underline{Y}(k-1) \underline{V}(k)^T \right] = E \left[\underline{V}(k) \underline{Y}(k-1)^T \right] = \underline{0}$$

Because of the assumption that $E \left[\underline{V}(k) \underline{V}(j)^T \right] = \underline{0}$ for $k \neq j$.

This result reduces Equation (2.23) to

$$\begin{aligned}\tilde{\underline{P}}(k/k) &= \underline{S}(k) \tilde{\underline{P}}(k-1/k-1) \underline{S}(k)^T \\ &\quad + \underline{G}(k) \underline{R}(k) \underline{G}(k)^T\end{aligned}\quad (2.31)$$

Equation (2.31) is the error covariance propagation equation which expresses the covariance of estimation error at time k in terms of the covariance of estimation error at time (k-1).

2. Initial conditions

To use Equations (2.19) and (2.31) one needs to know initial condition values.

The mean of estimation error at time k=0 is

$$\begin{aligned}\tilde{\mu}(0) &= E \left[\hat{\underline{X}}(0/0) - \underline{X}(0) \right] \\ &= E \left[\hat{\underline{X}}(0/0) \right] - \underline{X}(0)\end{aligned}$$

Using Equation (2.25) this becomes

$$\tilde{\mu}(0) = E \left\{ \left[\underline{I} - \underline{G}(0) \underline{C} \right] \hat{\underline{X}}(0/-1) + \underline{G}(0) \underline{h}(\underline{X}(0)) + \underline{G}(0) \underline{V}(0) \right\} - \underline{X}(0)$$

But

$$E \left[\hat{\underline{X}}(0/-1) \right] = \hat{\underline{X}}(0/-1)$$

$$E \left[\underline{G}(0) \underline{h}(\underline{X}(0)) \right] = \underline{G}(0) \underline{h}(\underline{x}(0))$$

and

$$E \left[\underline{G}(0) \underline{V}(0) \right] = 0$$

thus,

$$\begin{aligned}\tilde{\mu}(0) &= \left[\underline{I} - \underline{G}(0) \underline{C} \right] \hat{\underline{X}}(0/-1) - \underline{X}(0) \\ &\quad + \underline{G}(0) \underline{h}(\underline{x}(0))\end{aligned}\tag{2.32}$$

Equation (2.32) gives the initial condition for Equation (2.19).

The covariance of estimation error at time $k=0$ is

$$\tilde{P}(0/0) = E \left[\underline{Y}(0) \underline{Y}(0)^T \right] \quad (2.33)$$

Substituting Equation (2.26) into (2.33) gives

$$\begin{aligned} \tilde{P}(0/0) &= E \left[\underline{G}(0) \underline{V}(0) \underline{V}(0)^T \underline{G}(0)^T \right] \\ &= \underline{G}(0) \underline{R}(0) \underline{G}(0)^T \end{aligned} \quad (2.35)$$

Equation (2.35) gives the initial condition for equation (2.31).

3. Mean and covariance of prediction error

The one-step prediction is given by

$$\hat{\underline{X}}(k+1/k) = \phi \hat{\underline{X}}(k/k) + \underline{\Delta} \underline{U}(k) \quad (2.36)$$

and the N -step prediction, based on the estimate $\underline{X}(k/k)$ is

$$\hat{\underline{X}}_p(k+N/k) = \phi^N \underline{X}(k/k) + \sum_{i=0}^{N-1} \phi^{N-i-1} \underline{\Delta} \underline{U}(k+i) \quad (2.37)$$

Defining the deterministic matrix $\underline{A}(N)$ as

$$\underline{A}(N) = \sum_{i=0}^{N-1} \phi^{N-i-1} \underline{\Delta} \underline{U}(k+i) \quad (2.38)$$

then

$$\hat{\underline{X}}_p(k+N/k) = \phi^N \hat{\underline{X}}(k/k) + \underline{A}(N) \quad (2.39)$$

The N -step prediction error is

$$\begin{aligned} \tilde{\underline{X}}_p(k+N/k) &= \hat{\underline{X}}_p(k+N/k) - \underline{X}(k+N) \\ &= \phi^N \hat{\underline{X}}(k/k) + \underline{A}(N) - \underline{X}(k+N) \end{aligned} \quad (2.40)$$

The mean of the N-step prediction error is defined as

$$\begin{aligned}
 \tilde{\underline{\mu}}_p(k+N/k) &= E \left[\tilde{\underline{X}}_p(k+N/k) \right] \\
 &= E \left[\phi^N \hat{\underline{X}}(k/k) + \underline{A}(N) - \underline{X}(k+N) \right] \\
 &= \phi^N E \left[\hat{\underline{X}}(k/k) \right] + \underline{A}(N) - \underline{X}(k+N) \quad (2.41)
 \end{aligned}$$

But

$$\begin{aligned}
 \tilde{\underline{\mu}}(k) &= E \left[\hat{\underline{X}}(k/k) - \underline{X}(k) \right] \\
 &= E \left[\hat{\underline{X}}(k/k) \right] - \underline{X}(k)
 \end{aligned}$$

and

$$E \left[\hat{\underline{X}}(k/k) \right] = \tilde{\underline{\mu}}(k) + \underline{X}(k) \quad (2.42)$$

Substituting (2.42) into (2.41) gives

$$\tilde{\underline{\mu}}_p(k+N/k) = \phi^N \tilde{\underline{\mu}}(k) + \phi^N \underline{X}(k) + \underline{A}(N) - \underline{X}(k+N) \quad (2.43)$$

The covariance of N-step prediction error is

$$\begin{aligned}
 \tilde{\underline{P}}(k+N/k) &= E \left\{ \left[\tilde{\underline{X}}_p(k+N/k) - \tilde{\underline{\mu}}_p(k+N/k) \right] \cdot \left[\tilde{\underline{X}}_p(k+N/k) \right. \right. \\
 &\quad \left. \left. - \tilde{\underline{\mu}}_p(k+N/k) \right]^T \right\} \quad (2.44)
 \end{aligned}$$

Using (2.40) and (2.43) yields

$$\begin{aligned}
 \tilde{\underline{X}}_p(k+N/k) - \tilde{\underline{\mu}}_p(k+N/k) &= \phi^N \hat{\underline{X}}(k/k) + \cancel{\underline{A}(N)} - \cancel{\underline{X}(k+N)} \\
 &\quad - \phi^N \tilde{\underline{\mu}}(k) - \phi^N \underline{X}(k) - \cancel{\underline{A}(N)} \\
 &\quad + \cancel{\underline{X}(k+N)} \\
 &= \phi^N \left[\hat{\underline{X}}(k/k) - \underline{X}(k) - \tilde{\underline{\mu}}(k) \right] \\
 &= \phi^N \left[\tilde{\underline{X}}(k/k) - \tilde{\underline{\mu}}(k) \right] \quad (2.45)
 \end{aligned}$$

which, when substituted into (2.44) gives

$$\begin{aligned}\tilde{\mathbf{P}}(k+N/k) &= E \left\{ \phi^N \begin{bmatrix} \tilde{\mathbf{X}}(k/k) & -\tilde{\mathbf{U}}(k) \\ -\tilde{\mathbf{U}}(k) & \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{X}}(k/k) \\ -\tilde{\mathbf{U}}(k) \end{bmatrix}^T \begin{bmatrix} \phi^N \\ \end{bmatrix}^T \right\} \\ &= \phi^N \tilde{\mathbf{P}}(k/k) \begin{bmatrix} \phi^N \\ \end{bmatrix}^T\end{aligned}\quad (2.46)$$

Equations (2.43) and (2.46) give the mean and covariance of N-step prediction error based on the mean and covariance of estimation error at time k.

In practice, the measurement equations are often linear, i.e.

$$\underline{\mathbf{Z}}(k) = \underline{\mathbf{C}} \underline{\mathbf{X}}(k) + \underline{\mathbf{V}}(k) \quad (2.47)$$

Then, in the derivation of the analytical equations, $\underline{h}(\underline{\mathbf{X}}(k))$ can be replaced with $\underline{\mathbf{C}} \underline{\mathbf{X}}(k)$. This will change Equation (2.15) to

$$\underline{\mathbf{D}}(k) = \begin{bmatrix} \underline{\mathbf{I}} - \underline{\mathbf{G}}(k) \underline{\mathbf{C}} \end{bmatrix} \begin{bmatrix} \underline{\Delta} \underline{\mathbf{U}}(k-1) - \underline{\mathbf{X}}(k) \end{bmatrix} \quad (2.48)$$

and Equation (2.32) to

$$\tilde{\underline{\mathbf{U}}}(0) = \begin{bmatrix} \underline{\mathbf{I}} - \underline{\mathbf{G}}(0) \underline{\mathbf{C}} \end{bmatrix} \begin{bmatrix} \hat{\underline{\mathbf{X}}}(0/-1) - \underline{\mathbf{X}}(0) \end{bmatrix} \quad (2.49)$$

4. Summary of key equations

(1). Mean of estimation error

$$\tilde{\underline{\mathbf{U}}}(k) = \underline{\mathbf{S}}(k) \tilde{\underline{\mathbf{U}}}(k-1) + \underline{\mathbf{D}}(k) + \underline{\mathbf{S}}(k) \underline{\mathbf{X}}(k-1) \quad (2.19)$$

(2). Covariance of estimation error

$$\begin{aligned}\tilde{\mathbf{P}}(k/k) &= \underline{\mathbf{S}}(k) \tilde{\mathbf{P}}(k-1/k-1) \underline{\mathbf{S}}(k)^T \\ &\quad + \underline{\mathbf{G}}(k) \underline{\mathbf{R}}(k) \underline{\mathbf{G}}(k)^T\end{aligned}\quad (2.31)$$

where

$$\underline{\mathcal{S}}(k) = [\underline{\mathcal{I}} - \underline{\mathcal{G}}(k) \underline{\mathcal{C}}] \underline{\mathcal{Q}} \quad (2.14)$$

$$\begin{aligned} \underline{D}(k) = [\underline{\mathcal{I}} - \underline{\mathcal{G}}(k) \underline{\mathcal{C}}] \underline{\Delta} \underline{U}(k-1) \\ + \underline{\mathcal{G}}(k) \underline{h}(\underline{X}(k)) - \underline{X}(k) \end{aligned} \quad (2.15)$$

with initial conditions

$$\begin{aligned} \underline{\tilde{\mu}}(0) = [\underline{\mathcal{I}} - \underline{\mathcal{G}}(0) \underline{\mathcal{C}}] \hat{\underline{X}}(0/-1) - \underline{X}(0) \\ + \underline{\mathcal{G}}(0) \underline{h}(\underline{X}(0)) \end{aligned} \quad (2.32)$$

$$\underline{\tilde{P}}(0/0) = \underline{\mathcal{G}}(0) \underline{R}(0) \underline{\mathcal{G}}(0)^T \quad (2.35)$$

(3). Mean and covariance of N-step prediction error

$$\begin{aligned} \underline{\tilde{\mu}}_p(k+N/k) = \underline{\mathcal{Q}}^N \underline{\tilde{\mu}}(k) + \underline{\mathcal{Q}}^N \underline{X}(k) + \underline{A}(N) \\ - \underline{X}(k+N) \end{aligned} \quad (2.43)$$

$$\underline{\tilde{P}}(k+N/k) = \underline{\mathcal{Q}}^N \underline{\tilde{P}}(k/k) [\underline{\mathcal{Q}}^N]^T \quad (2.46)$$

where

$$\underline{A}(N) = \sum_{i=0}^{N-1} \underline{\mathcal{Q}}^{N-i-1} \underline{\Delta} \underline{U}(k+i) \quad (2.38)$$

If the system is linear and time-varying the $\underline{\mathcal{Q}}, \underline{\Delta}$

and $\underline{\mathcal{C}}$ matrices are time dependent and are denoted by $\underline{\mathcal{Q}}(k),$

$\underline{\Delta}(k), \underline{\mathcal{C}}(k).$ Since

$$\hat{\underline{X}}(k/k-1) = \underline{\mathcal{Q}}(k-1) \underline{X}(k-1/k-1) + \underline{\Delta}(k-1) \underline{U}(k-1) \quad (2.50)$$

and

$$\underline{Z}(k) = \underline{h}(\underline{X}(k)) + \underline{V}(k) \quad (2.51)$$

it is only necessary to replace the $\underline{\mathcal{Q}}$, $\underline{\Delta}$ and \underline{Q} matrices with $\underline{\mathcal{Q}}(k-1)$, $\underline{\Delta}(k-1)$ and $\underline{Q}(k)$ in Equations (2.14), (2.43), (2.15), (2.46) and (2.38).

The analytical equations derived in this chapter are applicable to linear filters with precomputed gain schedules.

From Equation (2.31) it is seen that the covariance of estimation error is independent of the track, however, the mean of estimation error is track dependent.

III. APPLICATION OF THE ANALYTICAL EQUATIONS TO THE CASE OF MORE THAN ONE TRACK

The equations derived in the previous chapter are based on one known track. Application of the analytical equations using one known track has been studied in Reference [1] and results have been tabulated.

If it is desired to evaluate the performance of a filter for various tracks, a different track can be used for each iteration (ensemble member) in a Monte-Carlo simulation. An alternative approach based on the previously derived analytical equations is:

(1). Apply the analytical equations to calculate the mean and covariance of estimation error for each track individually; these are the conditional means and covariances.

(2). Calculate the overall mean and covariance using the results of (1) and relationships involving the means of conditional expectation.

Assume that there are several tracks to be considered;

$$\begin{array}{cccc} (1) & (2) & (i) & (n) \\ \underline{X}(k), & \underline{X}(k), & \dots, & \underline{X}(k), \dots, \underline{X}(k) \end{array}$$

where each track has a given probability of occurrence

$$p_1, p_2, p_3, \dots, p_i, \dots, p_n \quad , \text{ i.e.}$$

$$p \left[\begin{array}{c} (i) \\ \underline{X}(k) = \underline{X}(k) \end{array} \right] = p_i$$

$$p_1 + p_2 + \dots + p_i + \dots + p_n = 1 \quad (3.1)$$

The mean $\tilde{\underline{u}}^{(i)}(k)$ and the covariance $\tilde{\underline{P}}^{(i)}(k/k)$ can be calculated for the i 'th track $\underline{X}(k)$. Since $\underline{X}(k)$ has been used for the calculation of the $\tilde{\underline{u}}^{(i)}(k)$, $\tilde{\underline{u}}^{(i)}(k)$ is the conditional expectation, i.e.

$$\tilde{\underline{u}}^{(i)}(k) = E \left[\tilde{\underline{X}}^{(i)}(k) / \underline{X}(k) = \underline{X}(k) \right] \quad (3.2)$$

Which is the mean of estimation error given the track $\underline{X}(k)$.

Thus, conditional expectations can be computed for each of the tracks.

To calculate the mean of estimation error, one needs a relationship between the conditional means and the mean.

A. CONDITIONAL EXPECTATION

The conditional expectation is defined as [2]:

$$E \left[\underline{A} / \underline{B} = \underline{b}_j \right] = \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \underline{a} f_A(\underline{a} / \underline{B} = \underline{b}_j) \cdot da_1 \cdot da_2 \dots da_n \quad (3.3)$$

Where

\underline{A} is a continuous random vector.

\underline{B} is a discrete random vector.

$f(\underline{a} / \underline{B} = \underline{b}_j)$ is the probability density function of the random vector \underline{A} given $\underline{B} = \underline{b}_j$.

The expected value of a continuous random vector \underline{A} is

$$E[\underline{A}] = \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \underline{a} f_A(\underline{a}) da_1 \cdot da_2 \cdot \dots da_n \quad (3.4)$$

Equation (3.4) can also be written in terms of the joint probability density function of continuous random vectors, i.e.

$$E[\underline{A}] = \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \underline{a} f_{AB}(\underline{a}, \underline{b}) \cdot da_1 \dots da_n \cdot db_1 \dots db_m \quad (3.5)$$

From probability theory the joint probability density function can be expressed in terms of conditional density function.

Since the random vector \underline{B} is discrete (point conditioning),

$$f_A(\underline{a} / \underline{B}=\underline{b}_j) = \frac{f_{AB}(\underline{a}, \underline{b})}{P[\underline{B}=\underline{b}_j]} \quad (3.6)$$

or

$$f_{AB}(\underline{a}, \underline{b}) = f_A(\underline{a} / \underline{B}=\underline{b}_j) \cdot P[\underline{B}=\underline{b}_j] \quad (3.7)$$

Substituting Equation (3.7) into (3.5) and replacing the integrals with a summation sign for the random vector \underline{B} (since \underline{B} is a discrete random vector) gives

$$E[\underline{A}] = \sum_j \left[\int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \underline{a} f_A(\underline{a} / \underline{B}=\underline{b}_j) \cdot da_1 \dots da_n \right] \cdot P[\underline{B}=\underline{b}_j] \quad (3.8)$$

The term inside the brackets in Equation (3.8) is the conditional expectation given by Equation (3.3). Thus, Equation (3.8) takes the final form of

$$E[A] = \sum_j E[A / B=b_j] \cdot P[B=b_j] \quad (3.9)$$

Equation (3.9) can be used to define the mean of estimation error in terms of individual means calculated for each track, i.e.

$$\begin{aligned} \tilde{\mu}(k) &= E[\tilde{X}(k)] \\ &= \sum_{i=1}^n E\left[\tilde{X}(k) / \overset{(i)}{X}(k)=\underline{X}(k)\right] \cdot P\left[\overset{(i)}{X}(k)=\underline{X}(k)\right] \quad (3.10) \end{aligned}$$

$$= \sum_{i=1}^n \tilde{\mu}^{(i)}(k) \cdot P_i \quad (3.11)$$

where

$$P_i = P\left[\overset{(i)}{X}(k)=\underline{X}(k)\right]$$

From Equation (2.31) it is seen that the covariance of estimation error is independent of the track for linear systems. So it can be calculated once for all tracks.

B. A LINEAR PROBLEM

An example of the application of the analytical equations to a simplified fire control estimation problem has been

presented in [1]. In this section, the same problem is considered for three tracks with given probabilities of occurrence. The simulation has been done using both the analytical equations (program 'EVAL') and a Monte-Carlo algorithm.

The problem is a part of a simplified anti-aircraft fire control estimation problem in one dimension. The filter is a Kalman filter which is derived by solving Equations (2.2) through (2.11) subject to the following assumptions. The model of target motion is a point mass moving with constant velocity. The state equations are

$$\underline{X}(k+1) = \underline{\phi} \underline{X}(k) + \underline{\Gamma} \underline{W}(k) \quad (3.12)$$

where

$$\underline{\phi} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad (3.13)$$

$$\underline{\Gamma} = \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix} \quad (3.14)$$

T is the time between measurements and equals 1 second.

The measurement equation is

$$\underline{Z}(k) = \underline{C} \underline{X}(k) + \underline{V}(k)$$

$$\underline{Z}(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} \underline{X}(k) + \underline{V}(k) \quad (3.15)$$

where

$X_1(k)$ is the range of the target.

$X_2(k)$ is the range rate of the target.

$V(k)$ is the measurement noise.

$w(k)$ is a random forcing input.

It is assumed that

$$E [v(k)] = 0 \quad (3.16)$$

$$\begin{aligned} E [v(k)^2] &= 625 \text{ (m)}^2 \\ &= R \end{aligned} \quad (3.17)$$

$$E [W(k)] = 0 \quad (3.18)$$

$$\begin{aligned} E [W(k)^2] &= 225 \text{ (m/sec}^2\text{)}^2 \\ &= Q \end{aligned} \quad (3.19)$$

The filter initial conditions are

$$E [\underline{X}(0)] = \begin{bmatrix} 50 \times 10^3 \text{ m} \\ -600 \text{ m/sec} \end{bmatrix} \quad (3.20)$$

$$= \hat{\underline{X}}(0/-1) \quad (3.21)$$

And the initial covariance matrix is

$$P(0/-1) = \begin{bmatrix} 10^9 & 0 \\ 0 & 10^9 \end{bmatrix} \quad (3.22)$$

Using this value for $P(0/-1)$ the gain and covariance

equations were solved to determine the gain schedule $\underline{G}(k)$,

$k = 0, 1, 2, \dots,$

In the simulations three tracks (true trajectories of the target) were used with initial conditions

$$\underline{X}^{(1)}(0) = \begin{bmatrix} 60 \times 10^3 \text{ m} \\ -600 \text{ m/sec} \end{bmatrix} \quad (3.23)$$

$$\underline{X}^{(2)}(0) = \begin{bmatrix} 55 \times 10^3 \text{ m} \\ -600 \text{ m/sec} \end{bmatrix} \quad (3.24)$$

$$\underline{X}^{(3)}(0) = \begin{bmatrix} 70 \times 10^3 \text{ m} \\ -600 \text{ m/sec} \end{bmatrix} \quad (3.25)$$

The probabilities of occurrence of the tracks are

$$P_1 = P[\underline{X}(k) = \underline{X}^{(1)}(k)] = 0.3 \quad (3.26)$$

$$P_2 = P[\underline{X}(k) = \underline{X}^{(2)}(k)] = 0.3 \quad (3.27)$$

$$P_3 = P[\underline{X}(k) = \underline{X}^{(3)}(k)] = 0.4 \quad (3.28)$$

Figures 3.3 through 3.6 show a comparison of the results obtained by using the analytical equations and Monte-Carlo simulation. Continuous curves represent Monte-Carlo results and triangles represent the results obtained using the analytical equations. 10,000 Monte-Carlo runs were used (3000 runs for the first track, 3000 runs for the second

track and 4000 runs for the third track). From Figures 3.3 through 3.6 it is seen that the two sets of results are very close -- especially the covariances of estimation error.

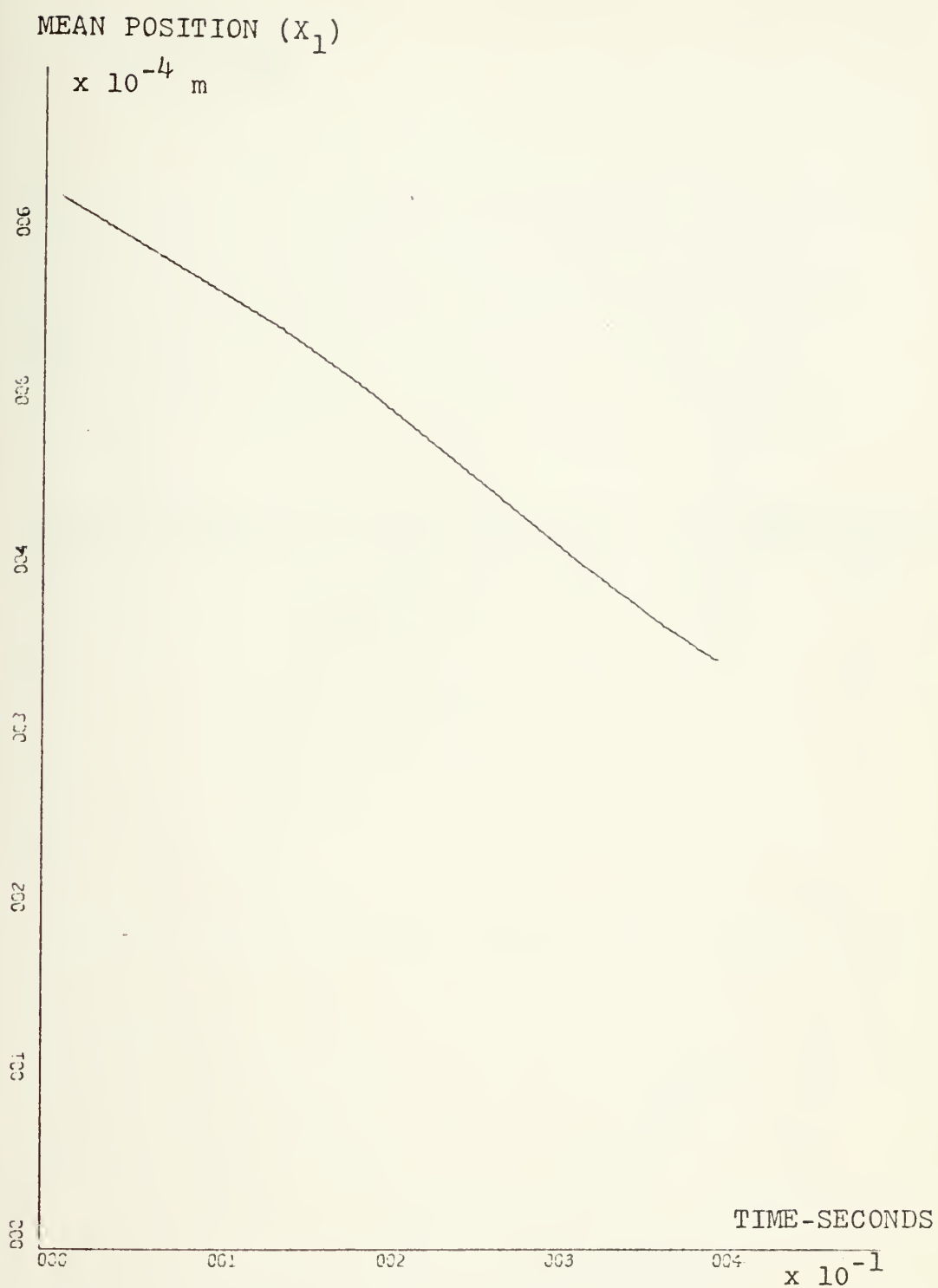


FIG. 3.1. Time history of mean position.

17 INCH.

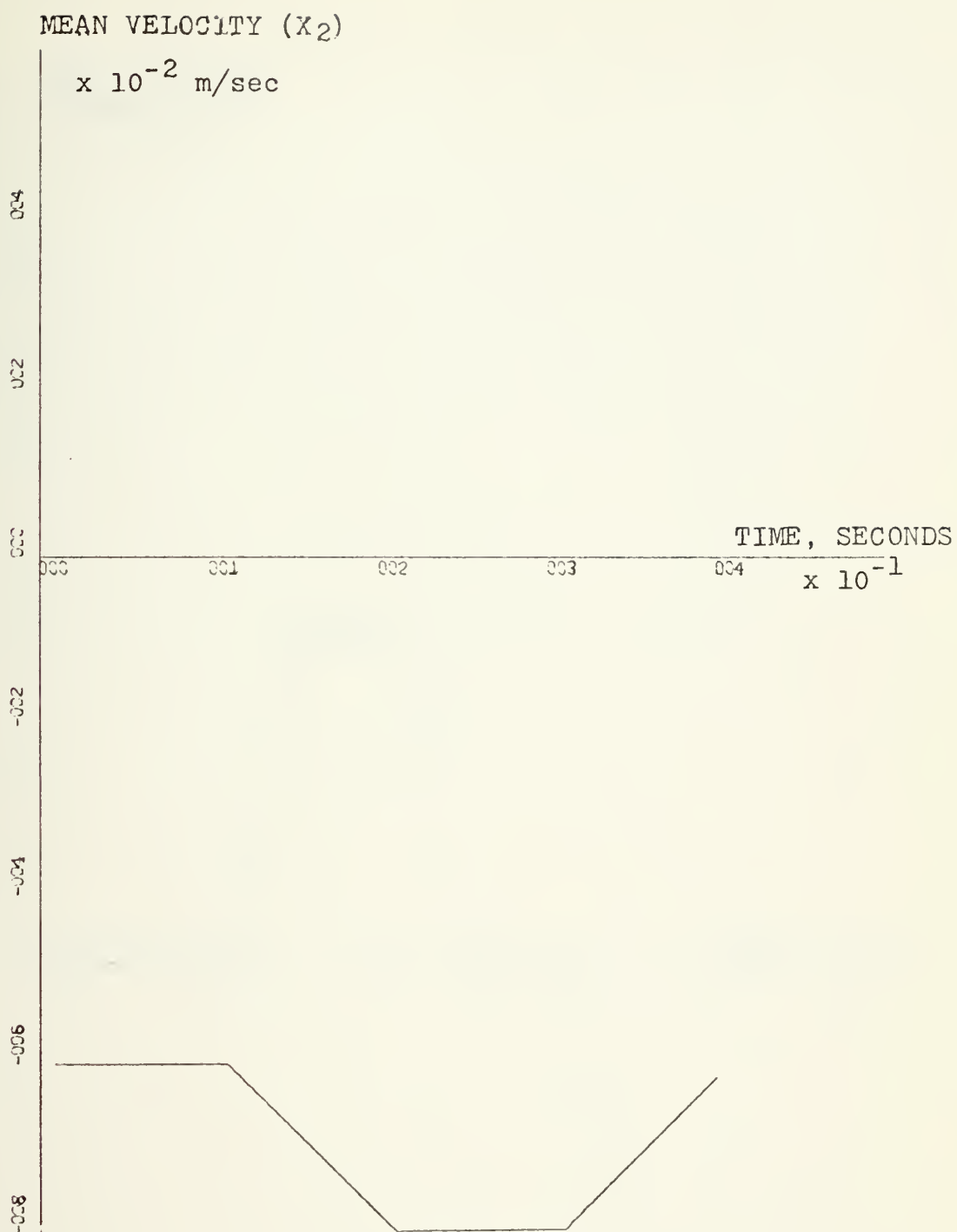


FIG. 3.2. Time history of mean velocity.

X-SCALE=1.00E+01 UNITS INCH.

Y-SCALE=2.00E+02 UNITS INCH.

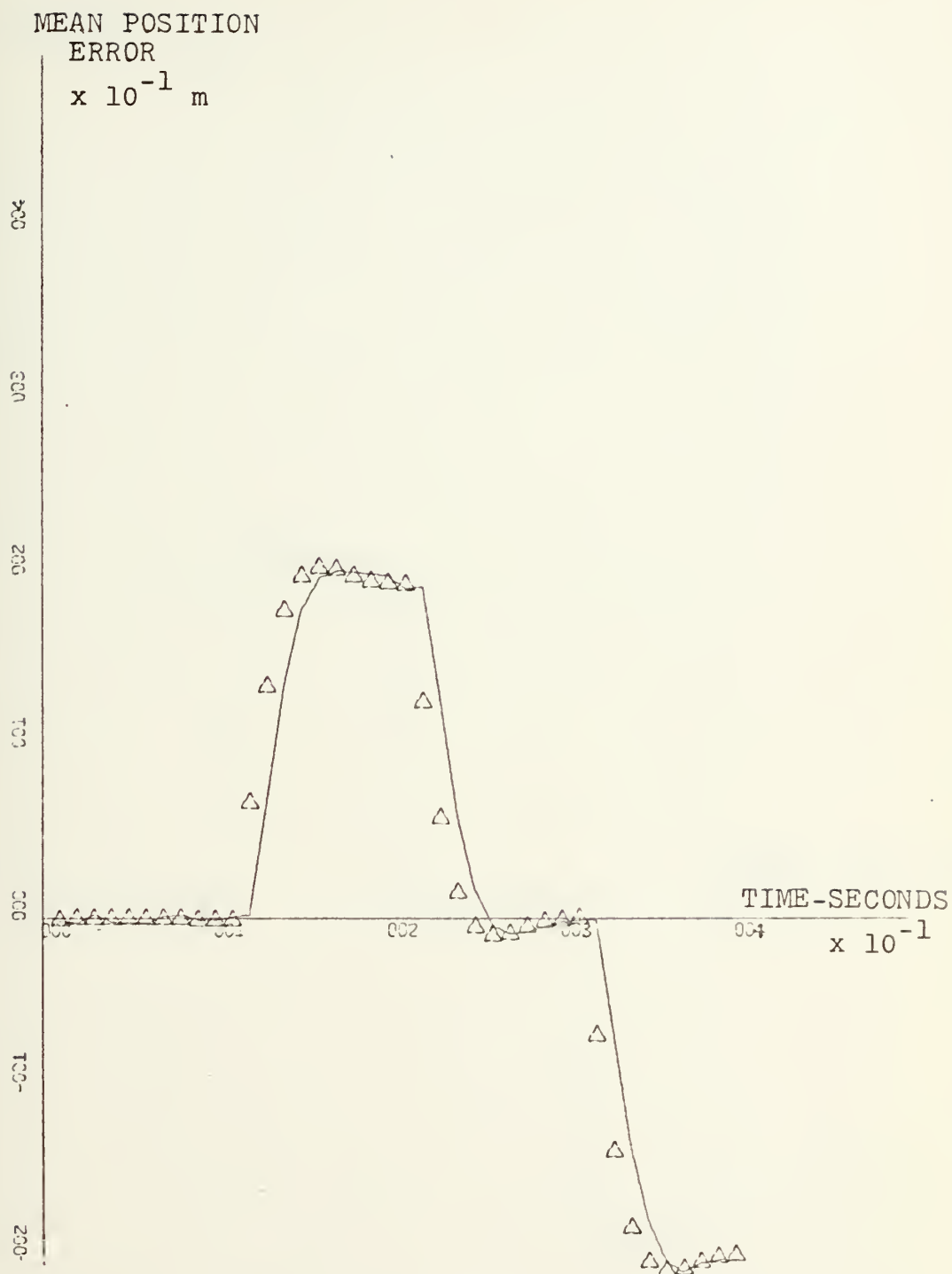


FIG. 3.3. Position estimation error history.

X-SCALE=1.00E+01 UNITS INCH.

Y-SCALE=1.00E+01 UNITS INCH.

MEAN VELOCITY
ERROR
x 10⁻¹ m/sec

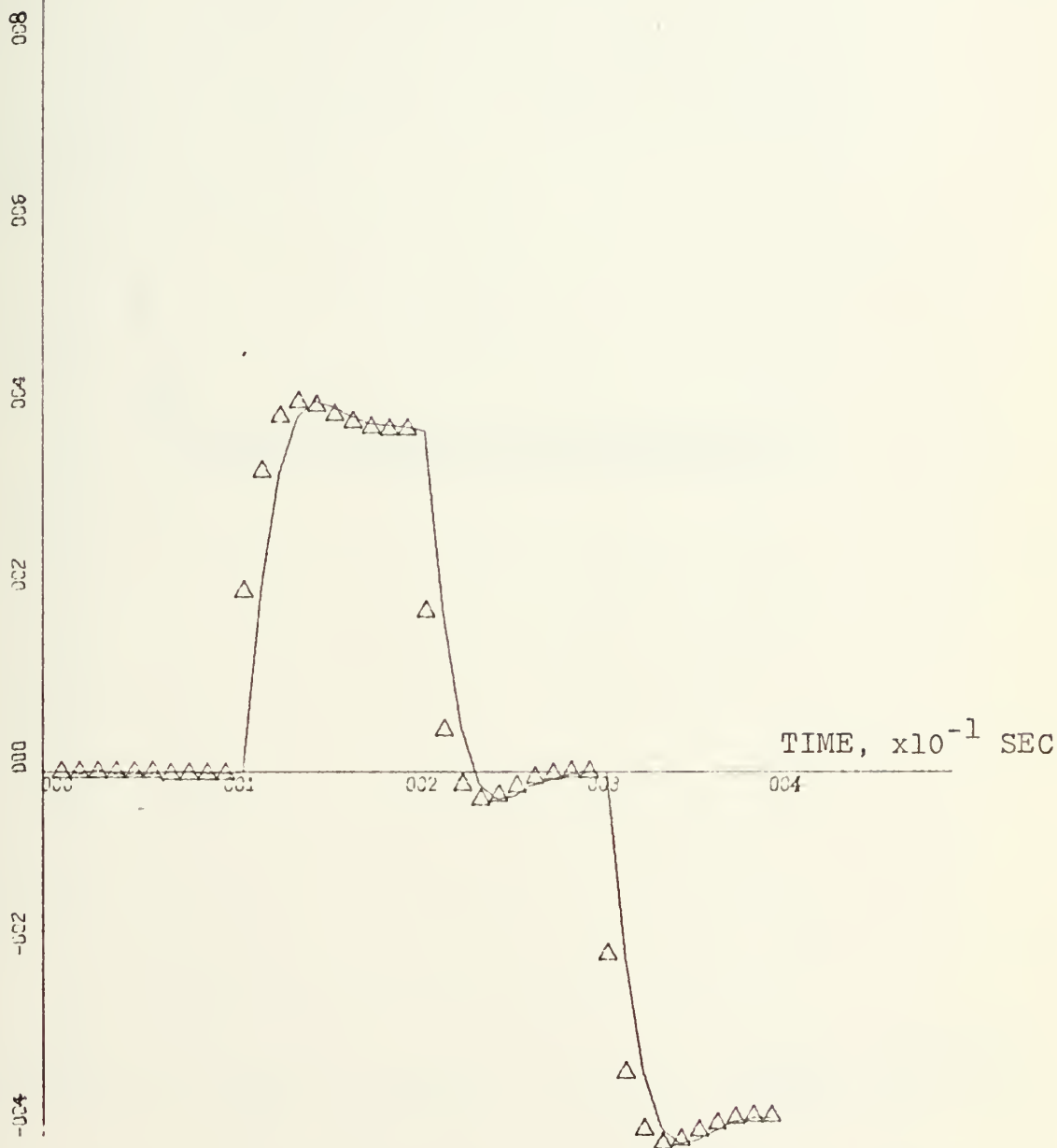


FIG. 3.4. Velocity estimation error history.

```
X-SCALE=1.00E+01 UNITS INCH.  
Y-SCALE=2.00E+01 UNITS INCH.
```

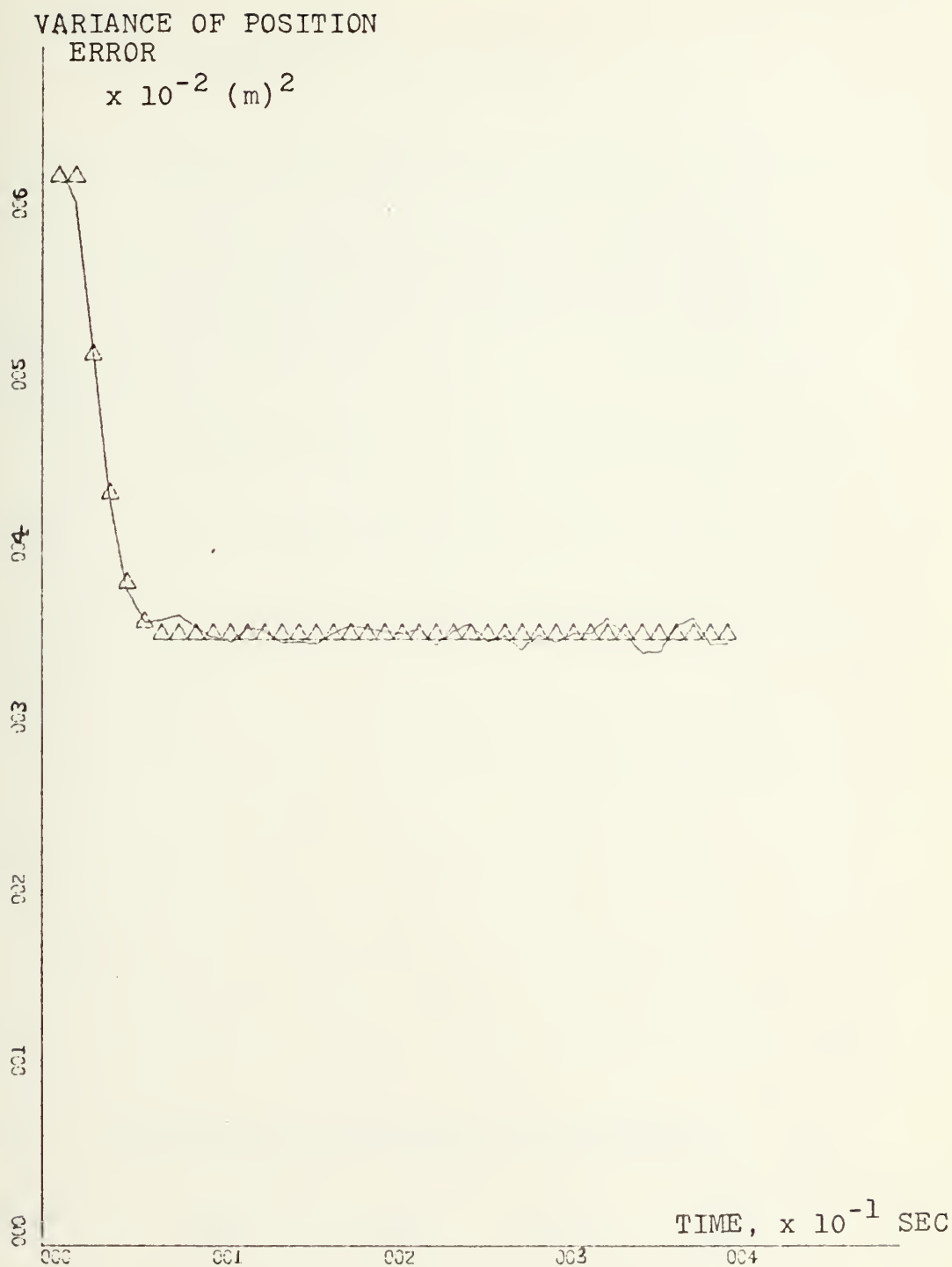



FIG. 3.5. Variance of position estimation error vs. time.

X-SCALE=1.00E+01 UNITS INCH.

Y-SCALE=1.00E+02 UNITS INCH.

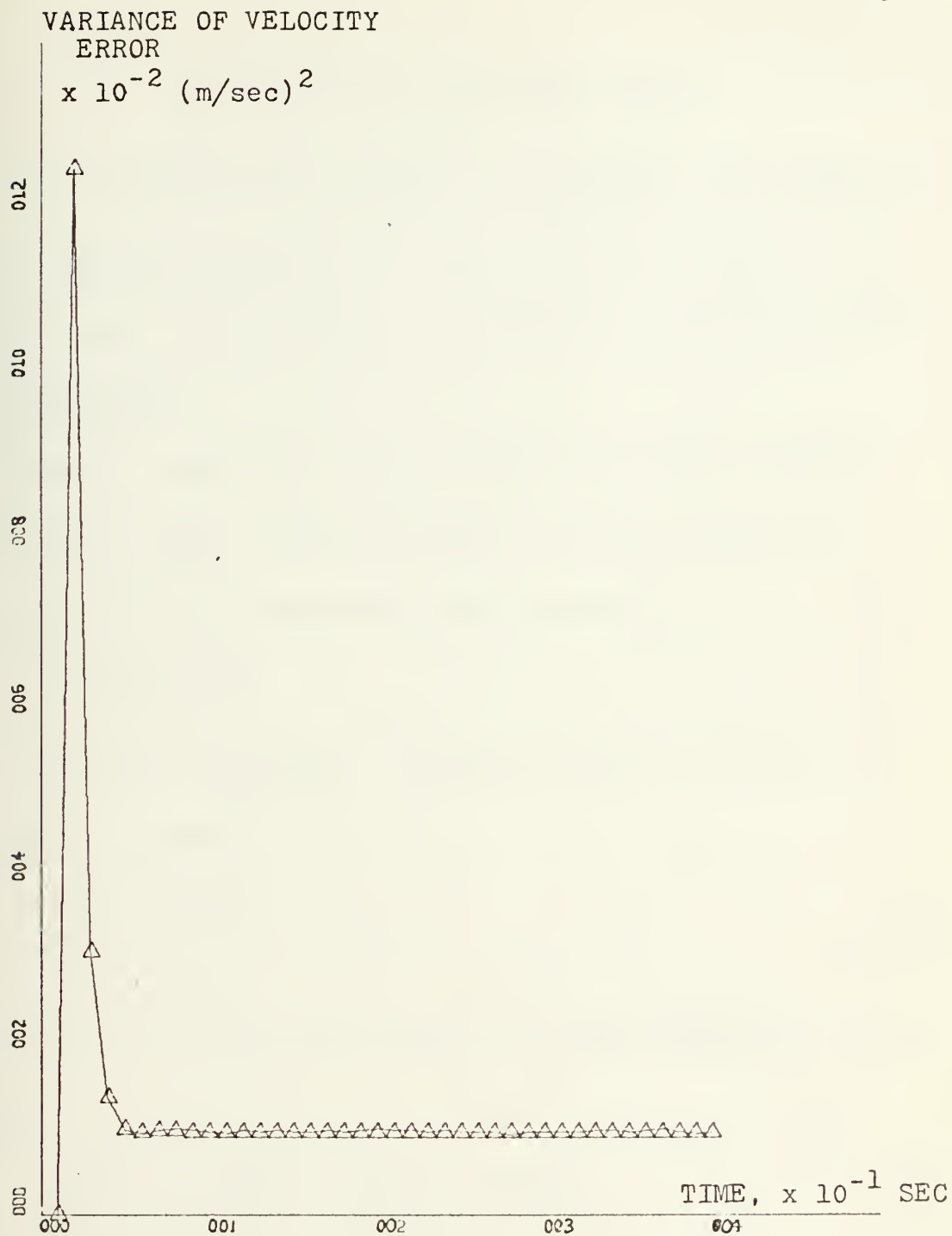


FIG. 3.6. Variance of velocity estimation error vs. time.

X-SCALE:-1.00E+01 UNITS INCH.

Y-SCALE:-2.00E+02 UNITS INCH.

IV. ESTIMATORS FOR NONLINEAR MODELS

The extended-Kalman filter is a commonly used approach for nonlinear estimation problems. In this chapter, the application of extended-Kalman filters to nonlinear models is summarized.

Given a continuous and nonlinear estimation problem, in order to apply linear discrete filtering theory the problem is first linearized and discretized.

A. DISCRETIZATION

Consider a nonlinear continuous system of state and observation equations given by

$$\underline{X}(t) = \underline{f}(\underline{X}(t), \underline{U}(t), t) \quad (4.1)$$

$$\underline{Z}(t) = \underline{h}(\underline{X}(t)) + \underline{V}(t) \quad (4.2)$$

The discrete-time representation of these equations has the form

$$\underline{X}(k+1) = \underline{a}(\underline{X}(k), \underline{U}(k), k) + \underline{W}(k) \quad (4.3)$$

$$\underline{Z}(k) = \underline{C}(\underline{X}(k)) + \underline{V}(k) \quad (4.4)$$

Equation (4.3) can be obtained from (4.1) by using the relationship

$$\begin{aligned}
 \underline{X}(k+1) = & \underline{X}(k) + (\Delta t) \cdot \underline{f}(\underline{X}(k), \underline{U}(k), k) \\
 & + \frac{1}{2} (\Delta t)^2 \left[\frac{\partial \underline{f}}{\partial \underline{X}} (\underline{X}(k), \underline{U}(k), k) \right. \\
 & \cdot \underline{f}(\underline{X}(k), \underline{U}(k), k) + \frac{\partial \underline{f}}{\partial \underline{U}} (\underline{X}(k), \underline{U}(k), k) \cdot \dot{\underline{U}}(k) \left. \right] \\
 & + \underline{W}(k)
 \end{aligned} \tag{4.5}$$

which is the Taylor series expansion of equation (4.1).

$\underline{W}(k)$ is a random input which may also be used to approximate the effect of the truncated higher-order terms.

B. LINEARIZATION

If there is available a nominal trajectory $\underline{X}(k)$ and the control $\underline{U}(k)$ is known, one can linearize equations (4.3) and (4.4) by expanding them in a Taylor series about the nominal trajectory $\underline{X}'(k)$. For the state equations this gives

$$\begin{aligned}
 \underline{X}(k+1) = & \underline{a}(\underline{X}'(k), \underline{U}(k), k) + \frac{\partial \underline{a}}{\partial \underline{X}} \bigg|_{(\underline{X}'(k), \underline{U}(k), k)} [\underline{X}(k) - \underline{X}'(k)] \\
 & + \underline{W}(k) + \text{Higher order terms (H.O.T)}
 \end{aligned} \tag{4.6}$$

Defining the matrix $\phi(k)$ as

$$\phi(k) = \frac{\partial \underline{a}}{\partial \underline{X}} \bigg|_{(\underline{X}'(k), \underline{U}(k), k)} \tag{4.7}$$

and truncating the higher-order terms, Equation (4.6)

reduces to

$$\begin{aligned}
 \underline{X}(k+1) = & \phi(k) \underline{X}(k) + \underline{a}(\underline{X}'(k), \underline{U}(k), k) \\
 & - \phi(k) \underline{X}'(k) + \underline{W}(k)
 \end{aligned} \tag{4.8}$$

Since $\underline{X}'(k)$ is a known quantity, the second and third terms of the right hand side of equation (4.8) are known and deterministic. Defining the vector $\underline{U}'(k)$ as

$$\underline{U}'(k) = \underline{a}(\underline{X}'(k), \underline{U}(k), k) - \phi(k) \underline{X}'(k) \quad (4.9)$$

Equation (4.8) becomes

$$\underline{X}(k+1) = \phi(k) \underline{X}(k) + \underline{U}'(k) + \underline{W}(k) \quad (4.10)$$

Applying the same procedure to Equation (4.4), the linearized form of the observation equation is

$$\underline{Z}(k) = \underline{H}(k) \underline{X}(k) + \underline{C}(\underline{X}'(k)) - \underline{H}(k) \underline{X}'(k) + \underline{V}(k) \quad (4.11)$$

where

$$\underline{H}(k) = \left. \frac{\partial \underline{C}}{\partial \underline{X}} \right|_{\underline{X}'(k)} \quad (4.12)$$

Defining the vector $\underline{B}(k)$ as

$$\underline{B}(k) = \underline{C}(\underline{X}'(k)) - \underline{H}(k) \underline{X}'(k) \quad (4.13)$$

yields

$$\underline{Z}(k) = \underline{H}(k) \underline{X}(k) + \underline{B}(k) + \underline{V}(k) \quad (4.14)$$

Equations (4.10) and (4.14) represent a linear time-varying model, $\underline{U}'(k)$ and $\underline{B}(k)$ represent bias terms resulting from the linearization process. The analytical equations can be applied to this model by replacing the $\underline{U}(k-1)$ term in (2.15) with the $\underline{U}'(k-1)$ term in Equation (2.37) and defining new measurement equations given by

$$\begin{aligned}\underline{Z}'(k) &= \underline{Z}(k) - \underline{B}(k) \\ &= \underline{H}(k) \underline{X}(k) + \underline{V}(k)\end{aligned}\tag{4.15}$$

C. THE EXTENDED-KALMAN FILTER

Consider a nonlinear discrete system of state and observation equations given by

$$\underline{X}(k+1) = \underline{f}(\underline{X}(k), k) + \underline{W}(k)\tag{4.16}$$

$$\underline{Z}(k) = \underline{C}(\underline{X}(k)) + \underline{V}(k)\tag{4.17}$$

In these equations \underline{f} and \underline{C} are nonlinear functions of the state variables $\underline{X}(k)$, $\underline{W}(k)$ is a random forcing input and $\underline{V}(k)$ is measurement noise with the usual assumptions (an uncorrelated, zero mean random processes).

$$E[\underline{W}(k) \underline{W}(j)^T] = Q(k) \delta_{kj}$$

$$E[\underline{V}(k) \underline{V}(j)^T] = R(k) \delta_{kj}$$

In order to apply the linear filter equations, Equations (4.16) and (4.17) are expanded about the nominal trajectory $\underline{X}'(k)$ if it is available. In practice, it is possible to have an idea about the target trajectory for some problems, like satellite orbit determination problems, in which case one may predict the target trajectory to be very close to the true trajectory and thus, be able to define a nominal trajectory for the purpose of linearization. In such cases the gain schedule can be computed before estimation, but in

other kinds of problems, it may be impossible to obtain a nominal trajectory. An alternative is to linearize the problem at each time point about the best estimates of the states currently available. In this case the gains can only be calculated for each sample as the estimates are available. This approach is called the extended-Kalman filter.

Using an estimate to evaluate the linearization Equation (4.16) gives

$$\underline{X}(k+1) = \underline{\phi}(k) \underline{X}(k) + \underline{W}(k) \quad (4.18)$$

where

$$\underline{\phi}(k) = \left. \frac{\partial \underline{f}}{\partial \underline{X}} \right|_{\hat{\underline{X}}(k/k)} \quad (4.19)$$

Similarly Equation (4.17) yields

$$\underline{Z}(k) = \underline{H}(k) \underline{X}(k) + \underline{V}(k) \quad (4.20)$$

where

$$\underline{H}(k) = \left. \frac{\partial \underline{C}}{\partial \underline{X}} \right|_{\hat{\underline{X}}(k/k-1)} \quad (4.21)$$

$\underline{\phi}(k)$ and $\underline{H}(k)$ are obtained as indicated in the linearization process discussed earlier and evaluated using the estimate as the nominal track. The filter estimation equation is

$$\hat{\underline{X}}(k/k) = \hat{\underline{X}}(k/k-1) + \underline{G}(k) \left[\underline{Z}(k) - \underline{C}(\underline{X}(k/k-1)) \right] \quad (4.22)$$

The prediction equation is

$$\hat{\underline{X}}(k/k-1) = \underline{f}(\hat{\underline{X}}(k-1/k-1), k-1) \quad (4.23)$$

The gains are obtained by using the relationship

$$\begin{aligned} \underline{G}(k) = \underline{P}(k/k-1) \underline{H}(k)^T & \left[\underline{H}(k) \underline{P}(k/k-1) \underline{H}(k)^T \right. \\ & \left. + \underline{R}(k) \right]^{-1} \end{aligned} \quad (4.24)$$

where

the theoretical covariance equation is

$$\underline{P}(k/k) = \left[\underline{I} - \underline{G}(k) \underline{H}(k) \right] \underline{P}(k/k-1) \quad (4.25)$$

The covariance propagation equation is

$$\underline{P}(k/k-1) = \underline{\phi}(k-1) \underline{P}(k-1/k-1) \underline{\phi}(k-1)^T + \underline{Q}(k-1) \quad (4.26)$$

and the initial conditions are

$$\begin{aligned} \hat{\underline{X}}(0/-1) &= E \left[\underline{X}(0) \right] \\ &= \bar{\underline{X}}_0 \\ \underline{P}(0/-1) &= \underline{P}_0 \end{aligned} \quad (4.27)$$

$$= E \left\{ \left[\underline{X}(0) - \bar{\underline{X}}_0 \right] \left[\underline{X}(0) - \bar{\underline{X}}_0 \right]^T \right\}$$

V. APPLICATION OF THE ANALYTICAL EQUATIONS TO EXTENDED-KALMAN FILTERS: A RE-ENTRY PROBLEM

In this chapter, the approximate evaluation of the performance of an extended-Kalman filter using the analytical equations is discussed. Since the analytical equations are based on linear estimators with pre-computed gain schedules, application of these equations to a problem with gains which are not pre-computed is an approximation.

In an extended-Kalman filter the gains are evaluated on-line using the estimates as they are produced. This means that the gains will vary from run-to-run even if the track is the same. This variation is caused by the differences in measurement noise which in turn effect the estimates. To use the analytical equation approach to evaluate an extended-Kalman filter it is assumed that the linearization which results in Equations (4.10) and (4.15) is performed using the true track. This leads to a pre-computed gain schedule that is used to approximate the extended-Kalman filter gain schedule.

A. THE RE-ENTRY PROBLEM

In order to compare the results of using the analytical equations and Monte-Carlo simulation, a particular problem was selected which contains significant nonlinearities in both the state and observation equations. This problem deals with estimation of the altitude, velocity and constant

ballistic coefficient of a vertically falling body. The measurements are taken at discrete time intervals of 1 second by a radar that measures range in the presence of (discrete) white gaussian noise. The geometry of the problem is illustrated in Figure 5.1. It is assumed that the body is falling vertically.

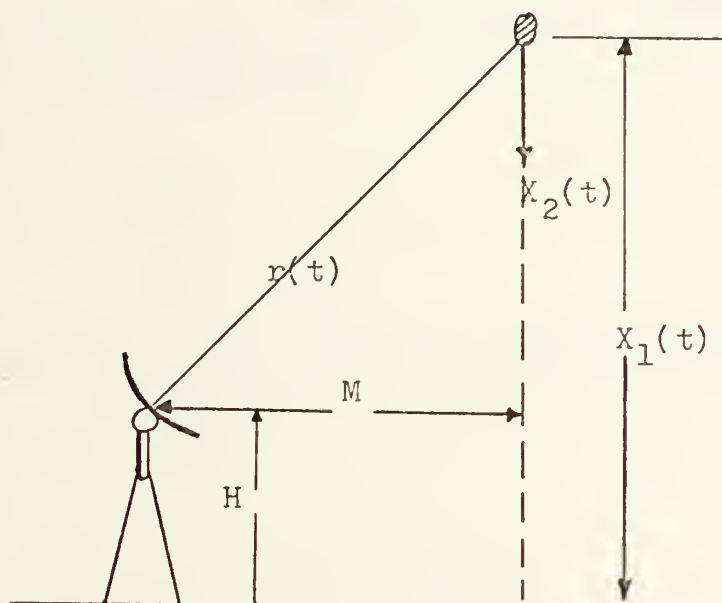


FIG. 5.1. Geometry of Re-entry problem.

The following definitions are used:

$X_1(t)$: Altitude
$X_2(t)$: Velocity (downward)
m	: Mass (constant)
C_D	: Drag coefficient (constant)
A	: Reference area for drag evaluation (constant)
ρ	: Mass density of the atmosphere
H	: Radar altitude
M	: Horizontal distance
$r(t)$: True range

It is assumed that the effect of gravity is negligible.

The equations of motion are

$$\begin{aligned}\dot{X}_1(t) &= -X_2(t) \\ \dot{X}_2(t) &= -\frac{C_D A \rho}{2 m} X_2(t)^2\end{aligned}\tag{5.1}$$

The air density is approximated by the exponential function

$$\rho = \rho_0 e^{-\gamma X_1(t)}\tag{5.2}$$

where

$$\gamma = 5 \times 10^{-5}\tag{5.3}$$

Defining

$$X_3(t) = \frac{C_D A \rho_0}{2 m}\tag{5.4}$$

which is a constant ballistic parameter, the state equations become

$$\dot{X}_1(t) = -X_2(t)$$

$$\dot{X}_2(t) = -X_2(t)^2 X_3(t) e^{-\gamma X_1(t)} \quad (5.5)$$

$$\dot{X}_3(t) = 0$$

which are of the form

$$\dot{\underline{X}}(t) = \underline{f}(\underline{X}(t)) \quad (5.6)$$

The measurement $r(t)$ (range) is given by

$$r(t) = \sqrt{M^2 + (X_1(t) - H)^2} \quad (5.7)$$

and is observed at discrete instants of time so that the observed sequence is

$$z(k) = \sqrt{M^2 + (X_1(k) - H)^2} + v(k) \quad (5.8)$$

where $v(k)$ is white gaussian noise with zero mean and constant variance

$$E[v(k)^2] = R$$

In this case, the output nonlinearity has the form

$$C(\underline{X}(k)) = \sqrt{M^2 + (X_1(k) - H)^2} \quad (5.9)$$

It is assumed that

$$H = 0$$

$$M = 100,000 \text{ ft}$$

and the filter initial conditions are

$$\begin{aligned}\hat{\underline{X}}(0/-1) &= \bar{\underline{X}}_0 \\ &= \begin{bmatrix} 3 \times 10^5 \text{ ft} \\ 2 \times 10^4 \text{ ft/sec} \\ 3 \times 10^{-5} \text{ ft}^{-1} \end{bmatrix}\end{aligned}\quad (5.10)$$

The assumed initial covariance matrix is

$$\underline{P}(0/-1) = \begin{bmatrix} 10^6 & 0 & 0 \\ 0 & 4 \times 10^6 & 0 \\ 0 & 0 & 10^{-4} \end{bmatrix}\quad (5.11)$$

and the variance of measurement noise is

$$R = 10 \times 10^4 \text{ (ft)}^2 \quad (5.12)$$

The true initial conditions of the falling body are

$$\underline{X}(0) = \begin{bmatrix} 3 \times 10^5 \text{ ft} \\ 2 \times 10^4 \text{ ft/sec} \\ 1 \times 10^{-3} \text{ ft}^{-1} \end{bmatrix}$$

Applying Equation (4.5) to Equation (5.5) it can be shown

that the discretized state equations are

$$\underline{X}(k+1) = \underline{a}(\underline{X}(k)) \quad (5.14)$$

$$X_1(k+1) = X_1(k) - X_2(k) + \frac{1}{2} X_2(k)^2 X_3(k) e^{-\gamma X_1(k)} \quad (5.15)$$

$$\begin{aligned}X_2(k+1) &= X_2(k) - X_2(k)^2 X_3(k) e^{-\gamma X_1(k)} - \frac{1}{2} X_2(k)^3 X_3(k) \\ &\quad \cdot e^{-\gamma X_1(k)} + X_2(k)^3 X_3(k)^2 e^{-2\gamma X_1(k)}\end{aligned}\quad (5.16)$$

$$x_3(k+1) = x_3(k) \quad (5.17)$$

Linearizing Equations (5.15) - (5.17) it can be shown that the $\underline{g}(k)$ matrix is

$$\underline{g}(k) = \frac{\partial \underline{a}}{\partial \underline{X}} \bigg|_{\underline{X}'(k)} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \quad (5.18)$$

where $\underline{X}'(k)$ is the nominal trajectory and the A_{ij} 's are

$$A_{11} = 1 - (1/2) x_2'(k)^2 x_3'(k) e^{-\gamma x_1'(k)} \quad (5.19)$$

$$A_{12} = -1 + x_2'(k) x_3'(k) e^{-\gamma x_1'(k)} \quad (5.20)$$

$$A_{13} = \frac{1}{2} x_2'(k)^2 e^{-\gamma x_1'(k)} \quad (5.21)$$

$$A_{21} = x_2'(k)^2 x_3'(k) e^{-\gamma x_1'(k)} + \frac{1}{2} \gamma^2 x_2'(k)^3 x_3'(k) e^{-\gamma x_1'(k)} - 2 \gamma x_2'(k)^3 x_3'(k)^2 e^{-2\gamma x_1'(k)} \quad (5.22)$$

$$A_{22} = 1 - 2 x_2'(k) x_3'(k) e^{-\gamma x_1'(k)} - \frac{3}{2} \gamma x_2'(k)^2 x_3'(k) e^{-\gamma x_1'(k)} + 3 x_2'(k)^2 x_3'(k)^2 e^{-2\gamma x_1'(k)} \quad (5.23)$$

$$A_{23} = -X_2'(k)^2 e^{\gamma X_1'(k)} - \frac{1}{2} \gamma X_2'(k)^3 e^{-\gamma X_1'(k)} + 2 X_2'(k)^3 X_3'(k) e^{-2 \gamma X_1'(k)} \quad (5.24)$$

$$A_{31} = 0 \quad (5.25)$$

$$A_{32} = 0 \quad (5.26)$$

$$A_{33} = 1 \quad (5.27)$$

The $\underline{H}(k)$ matrix is

$$\underline{H}(k) = \frac{\partial \underline{C}}{\partial \underline{X}} \bigg|_{\underline{X}'(k)} = \begin{bmatrix} X_1'(k) - H & 0 & 0 \\ \sqrt{M^2 + (X_1'(k) - H)^2} & 0 & 0 \end{bmatrix} \quad (5.28)$$

In the simulation of the problem, the true track was generated

by using the discretized state equations (5.15) through

(5.17). The linearization is made about the track in the

application of the analytical equations and $\underline{U}'(k)$ is defined

as

$$\underline{U}'(k) = \underline{a}(\underline{X}'(k), k) - \underline{g}(k) \underline{X}'(k) \quad (5.29)$$

If the true track is generated by solving the state equations then

$$\underline{X}'(k+1) = \underline{a}(\underline{X}'(k), k) \quad (5.30)$$

and

$$\underline{U}'(k) = \underline{X}'(k+1) - \underline{\varrho}(k) \underline{X}'(k) \quad (5.31)$$

$$\underline{U}'(k-1) = \underline{X}'(k) - \underline{\varrho}(k-1) \underline{X}'(k-1) \quad (5.32)$$

If one uses a track other than that generated by solving the state equations, then Equation (5.29) must be used for $\underline{U}'(k)$. The matrices $\underline{\varrho}(k)$, $\underline{H}(k)$ and the vector $\underline{U}'(k-1)$ can be used for application of the analytical equations. In the Monte-Carlo simulation of the extended-Kalman filter, one must replace $\underline{X}'(k)$ with the estimates $\hat{\underline{X}}(k/k)$ in Equations (5.19) through (5.27) and by $\hat{\underline{X}}(k/k-1)$ in Equation (5.28) ($\underline{\varrho}(k)$ must be evaluated about $\hat{\underline{X}}(k/k)$ and $\underline{H}(k)$ must be evaluated about $\hat{\underline{X}}(k/k-1)$).

Figures 5.5 through 5.10 illustrate the results of the analytical equations and the Monte-Carlo simulation of the extended-Kalman filter. The continuous curves represent the Monte-Carlo results (1000 runs) and the triangles represent results from the analytical equations. The figures show that the analytical equations have predicted better performance than that predicted by the Monte-Carlo simulation. Actually, this observation is not generally true. The difference between results of the two methods



FIG. 5.2. Altitude history of re-entry vehicle.

X-SCALE=5.00E+00 UNITS INCH.

Y-SCALE=5.00E+04 UNITS INCH.

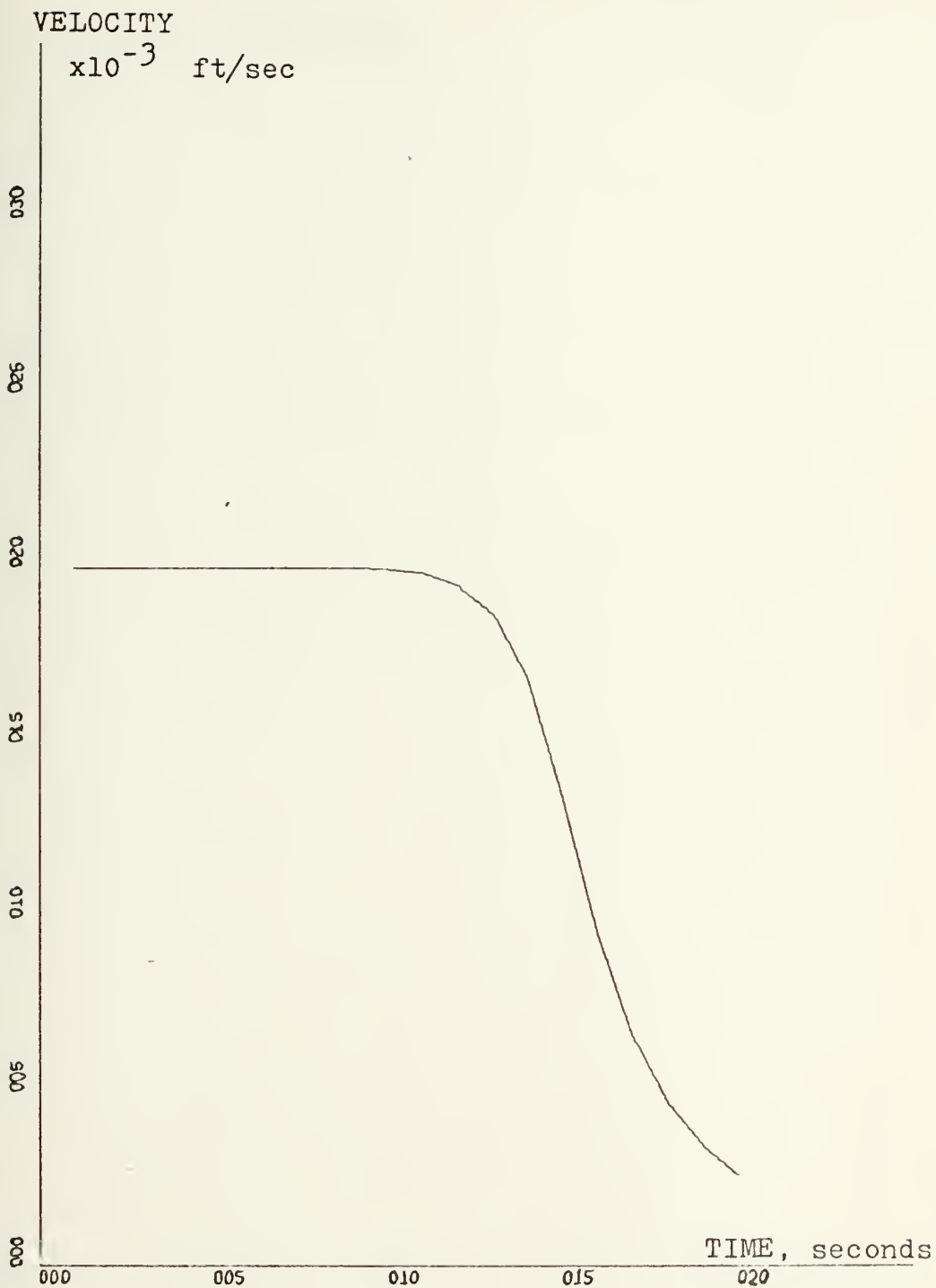


FIG. 5.3. Velocity history of re-entry vehicle.

X-SCALE=5.00E+00 UNITS INCH.

Y-SCALE=5.00E+03 UNITS INCH.



FIG. 5.4. Time history of ballistic coefficient.

X-SCALE=5.00E+00 UNITS INCH.

Y-SCALE=2.00E-04 UNITS INCH.

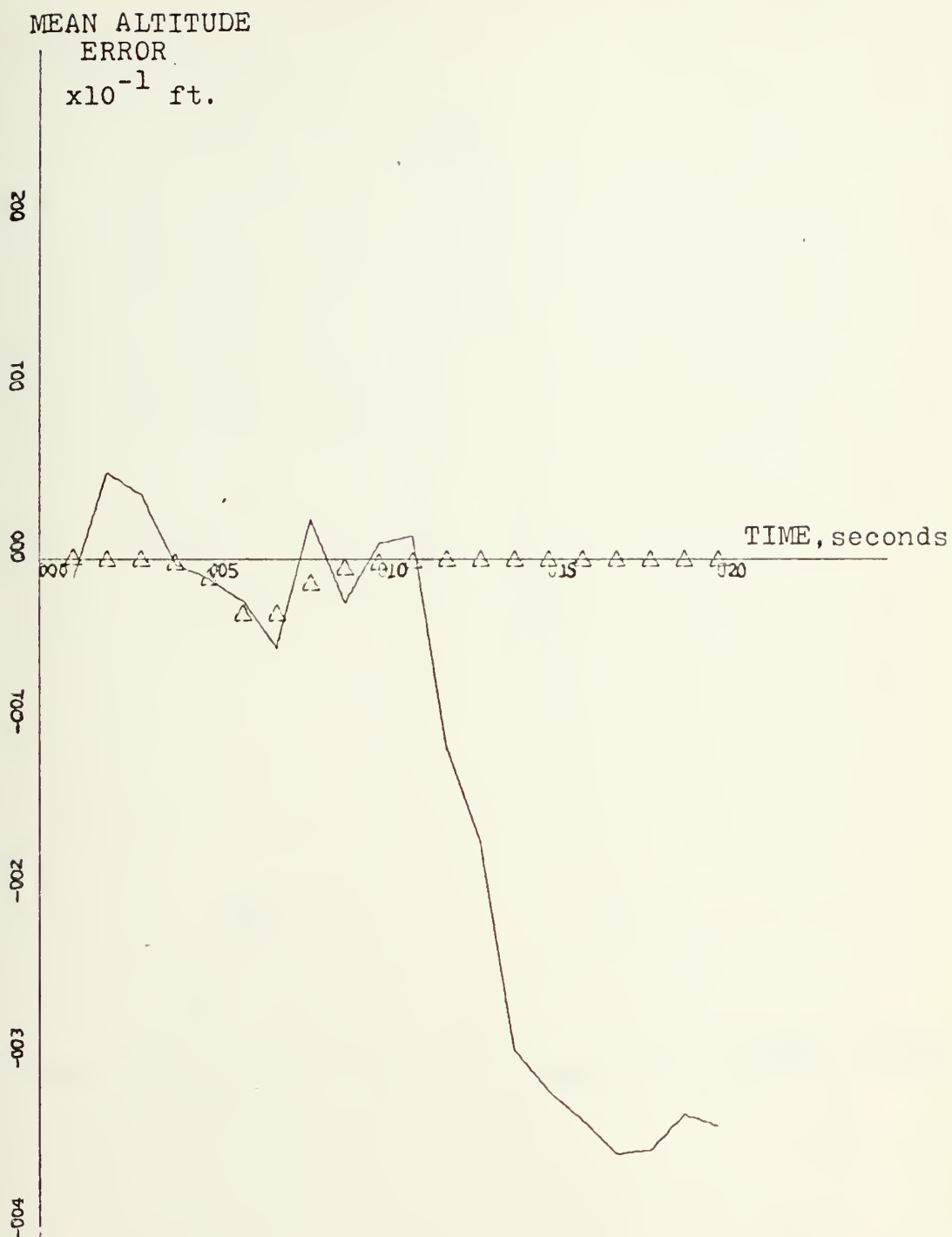


FIG. 5.5. Time history of altitude estimation error.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=1.00E+01 UNITS INCH.

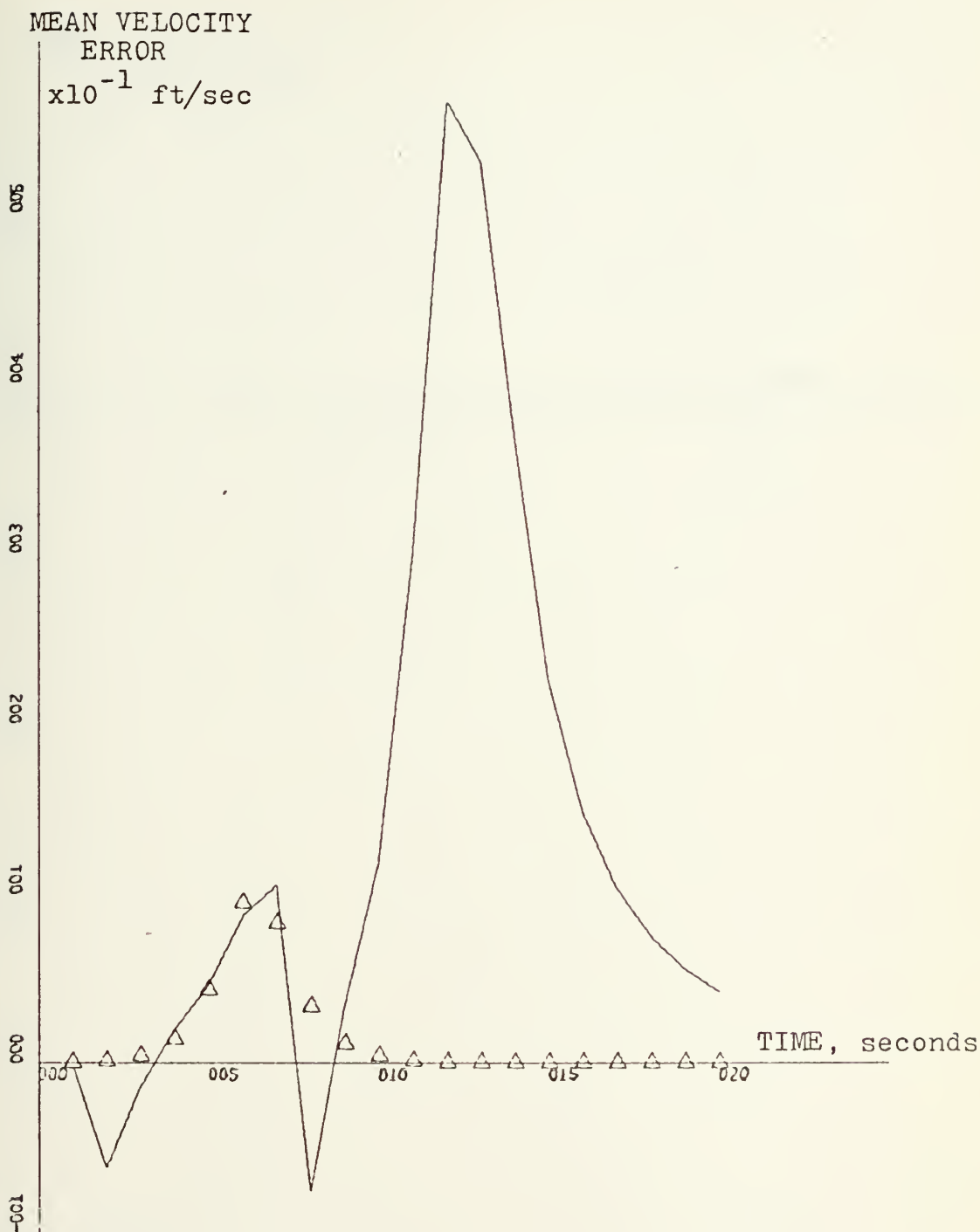


FIG. 5.6. Time history of the velocity estimation error.

X-SCALE=5.00E+00 UNITS INCH.

Y-SCALE=1.00E+01 UNITS INCH.

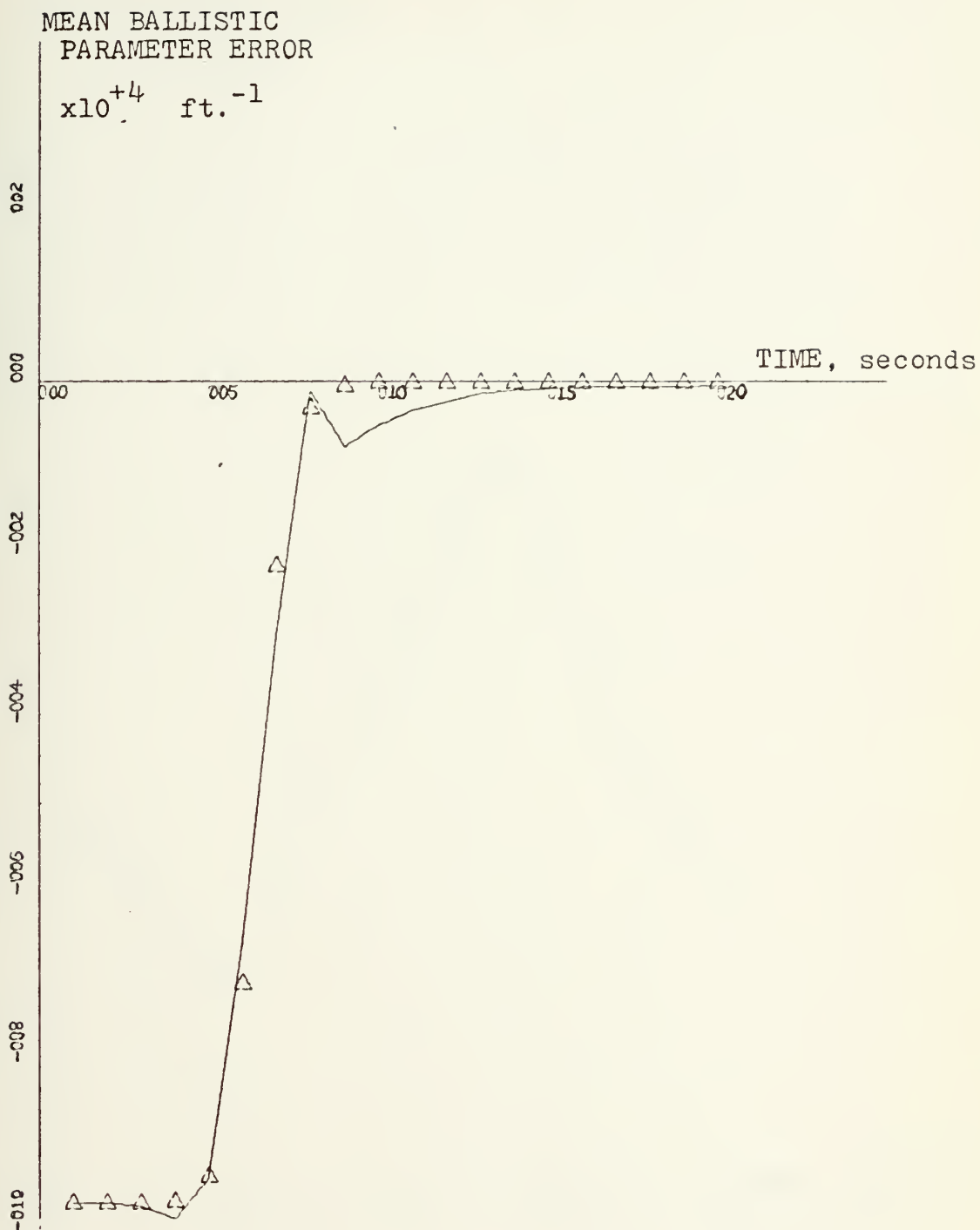


FIG. 5.7. Time history of the mean ballistic parameter estimation error.

X-SCALE: 5.00E+00 UNITS INCH.

Y-SCALE: 2.00E-04 UNITS INCH.

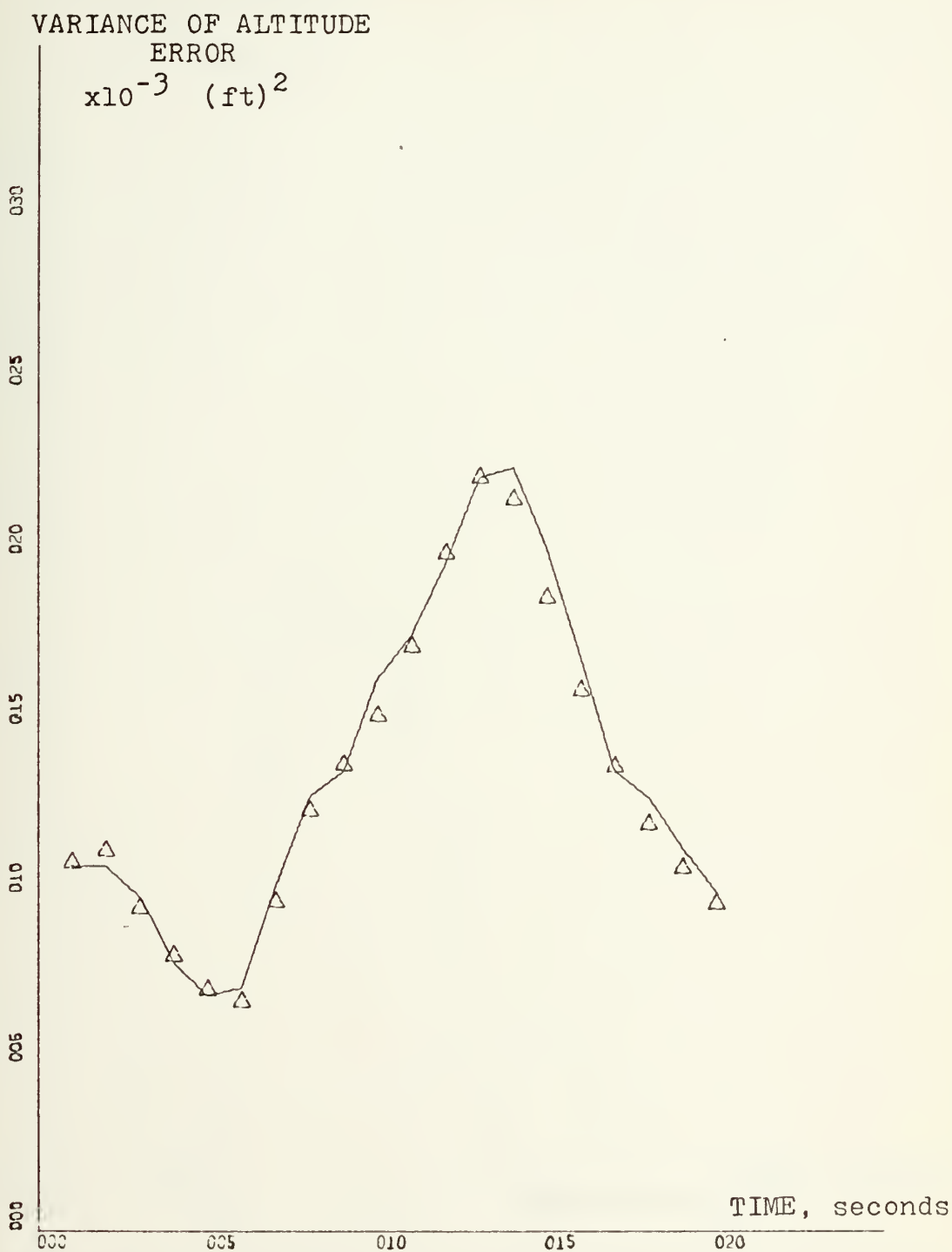


FIG. 5.8. Variance of altitude estimation error vs. time.

X-SCALE:-5.00E+00 UNITS INCH.

Y-SCALE:-5.00E+03 UNITS INCH.

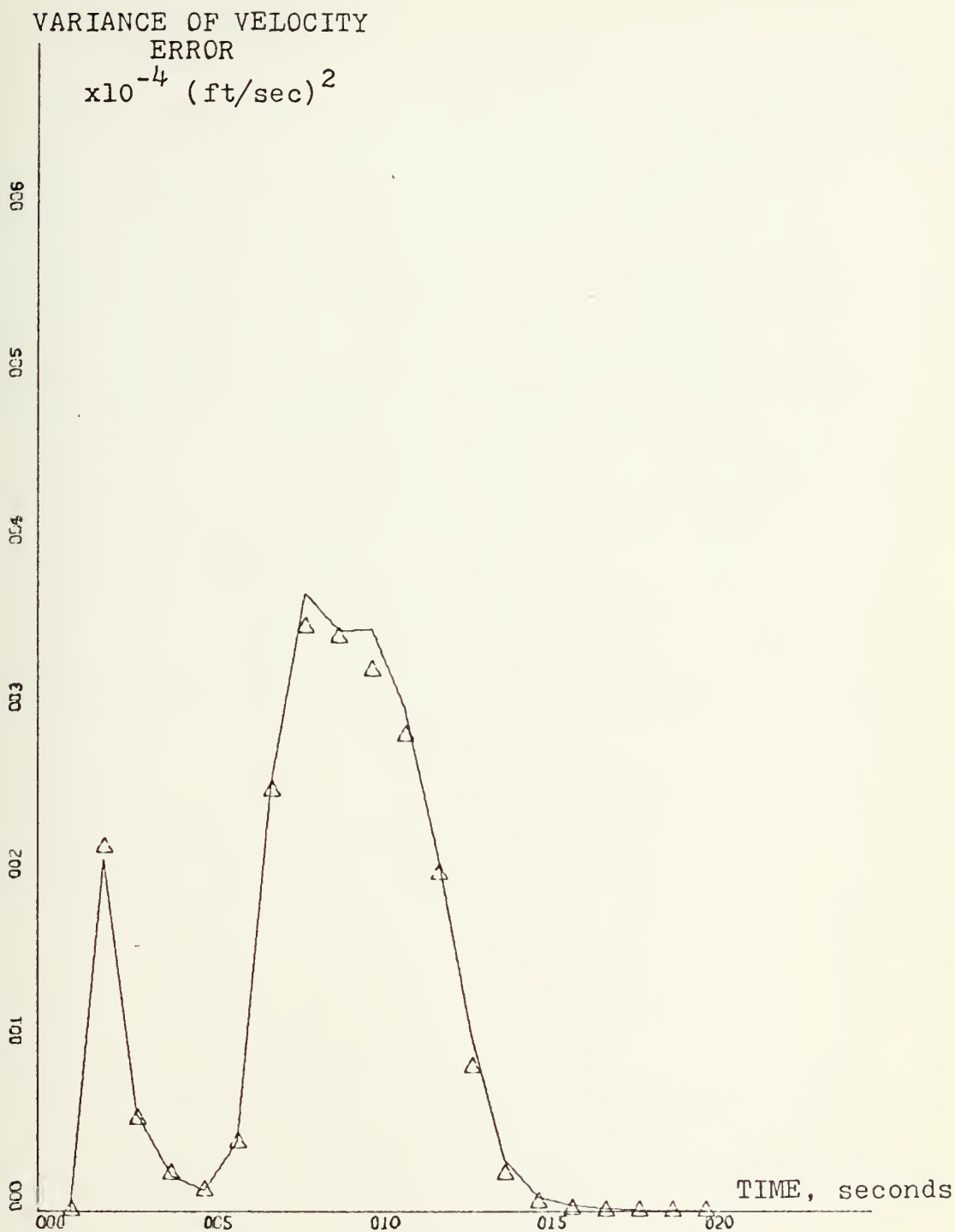


FIG. 5.9. Variance of velocity estimation error vs. time.

X-SCALE=5.00E+00 UNITS INCH.

Y-SCALE=1.00E+04 UNITS INCH.

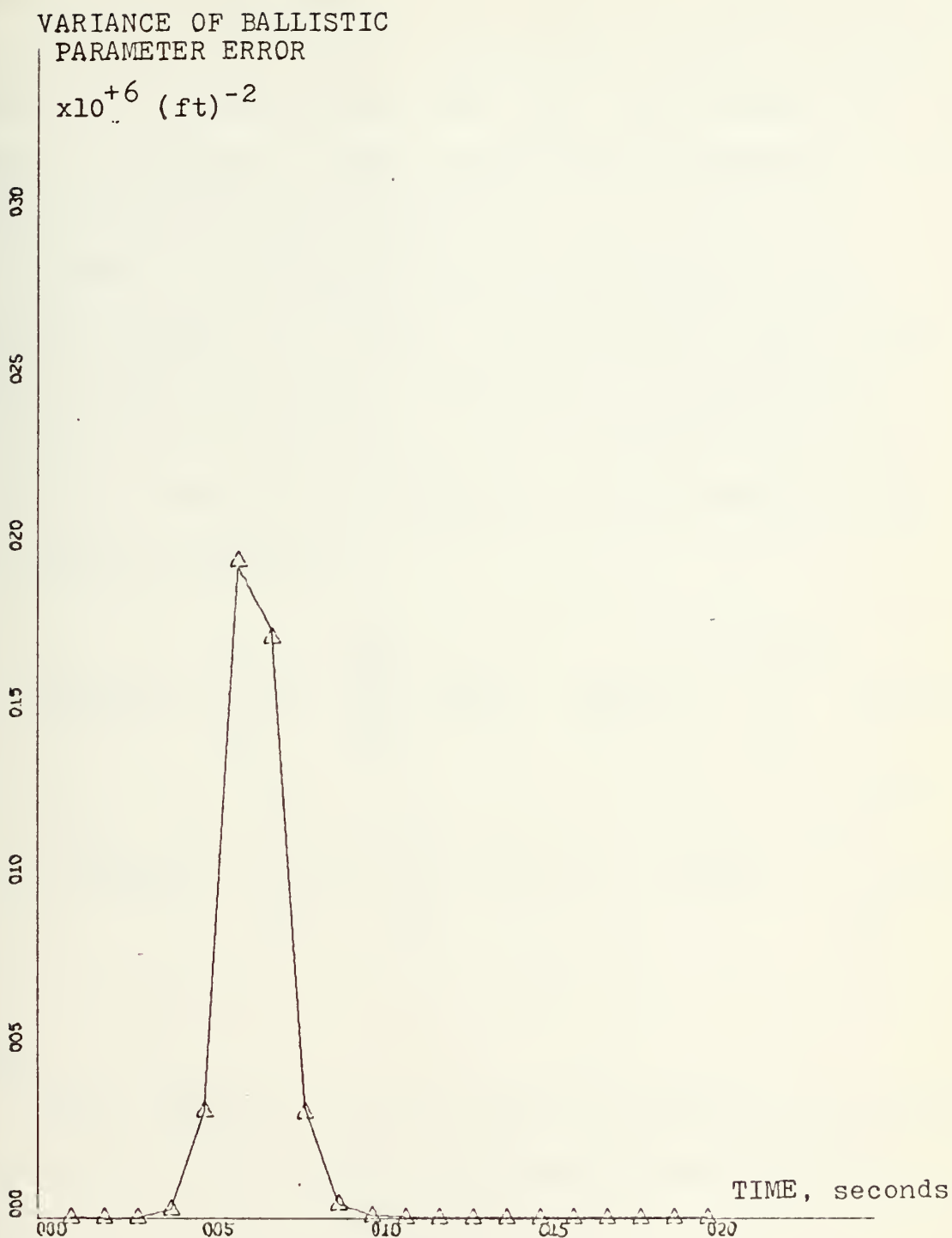


FIG. 5.10. Variance of ballistic parameter estimation error vs. time.

X-SCALE=5.00E+00 UNITS INCH.

Y-SCALE=5.00E-06 UNITS INCH.

depends on the measurement noise, the nature of the nonlinearities and the filter initial conditions. For the re-entry problem the two results are very close. For example, at 50,000 ft altitude, the mean altitude error difference between the two results is 30 feet.

In order to investigate the noise dependence of the difference between two results, the problem has been solved by each method for various covariances of the measurement noise and the root mean square of the difference has been calculated. The root mean square of the difference between two results is defined as

$$\Delta \tilde{\mu}_{\text{rms}}(I) = \left\{ \frac{1}{N} \sum_{k=1}^N \left[\tilde{\mu}_1(k) - \tilde{\mu}_2(k) \right]^2 \right\}^{1/2} \quad (5.33)$$

$$\Delta \underline{\text{VAR}}_{\text{rms}}(I) = \left\{ \frac{1}{N} \sum_{k=1}^N \left[\underline{\text{VAR}}_1(k) - \underline{\text{VAR}}_2(k) \right]^2 \right\}^{1/2} \quad (5.34)$$

where

$\tilde{\mu}_1(k)$, $\underline{\text{VAR}}_1(k)$ are the mean and variance of estimation error calculated by the Monte-Carlo simulation,

$\tilde{\mu}_2(k)$, $\underline{\text{VAR}}_2(k)$ are the mean and variance of estimation error calculated by the analytical equations,

N is the number of time points, and

$\Delta \tilde{\mu}_{\text{rms}} (I), \Delta \underline{\text{VAR}}_{\text{rms}} (I)$ are the rms values of the difference between the two results.

The results obtained have been tabulated in Table 5.1. From, Table 5.1 it is seen that the root mean square differences between the two results for both altitude and velocity estimation error increase as the covariance of measurement noise increases. But this is not true for X_3 which is the ballistic coefficient. Table 5.1 indicates that estimation of the ballistic coefficient is relatively insensitive to the measurement noise level.

Figures 5.11 through 5.16 represent the results of 50 Monte-Carlo runs (continuous curves) and the analytical equations. It is seen that there is significant difference between 1000 Monte-Carlo run-results and 50 Monte-Carlo run-results. (See Figure 5.5-5.10).

Variance of measurement noise	$\Delta\mu_{rms}(1)$ (feet)	$\Delta\mu_{rms}(2)$ (feet/sec)	$\Delta\mu_{rms}(3)$ (1/feet)	$\Delta VAR_{rms}(1)$ (feet) ²	$\Delta VAR_{rms}(2)$ (feet/sec) ²	$\Delta VAR_{rms}(3)$ (1/feet) ²
1	8.15043x10 ⁻²	4.1566x10 ⁻²	4.04867x10 ⁻⁵	5.2617x10 ⁻²	0.11749	3.0333x10 ⁻⁸
100	0.34382	0.5678	4.2500 x10 ⁻⁵	4.8689	7.2385	1.2267x10 ⁻⁷
1000	2.5657	3.5499	2.1497x10 ⁻⁵	47.8941	73.7134	5.0683x10 ⁻⁸
10,000	20.021	21.7225	3.1718x10 ⁻⁵	616.742	989.287	7.0329x10 ⁻⁸

TABLE 5.1. RMS deviations of means and variances obtained by Monte-Carlo and Analytical methods for various variances of measurement noise.

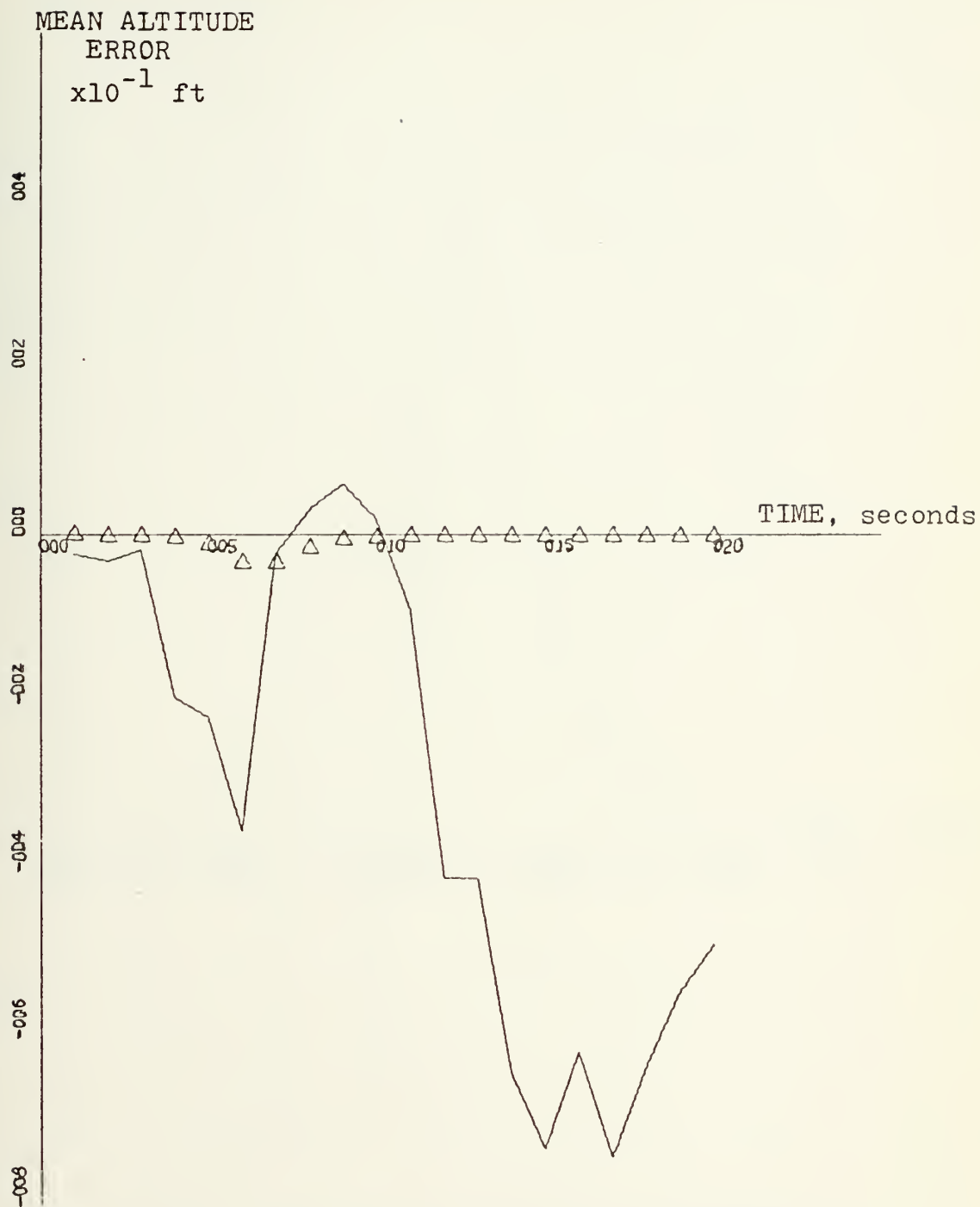


FIG. 5.11. Time history of the mean altitude estimation error for 50 Monte-Carlo runs.

X-SCALE=5.00E+00 UNITS INCH.

Y-SCALE=2.00E+01 UNITS INCH.

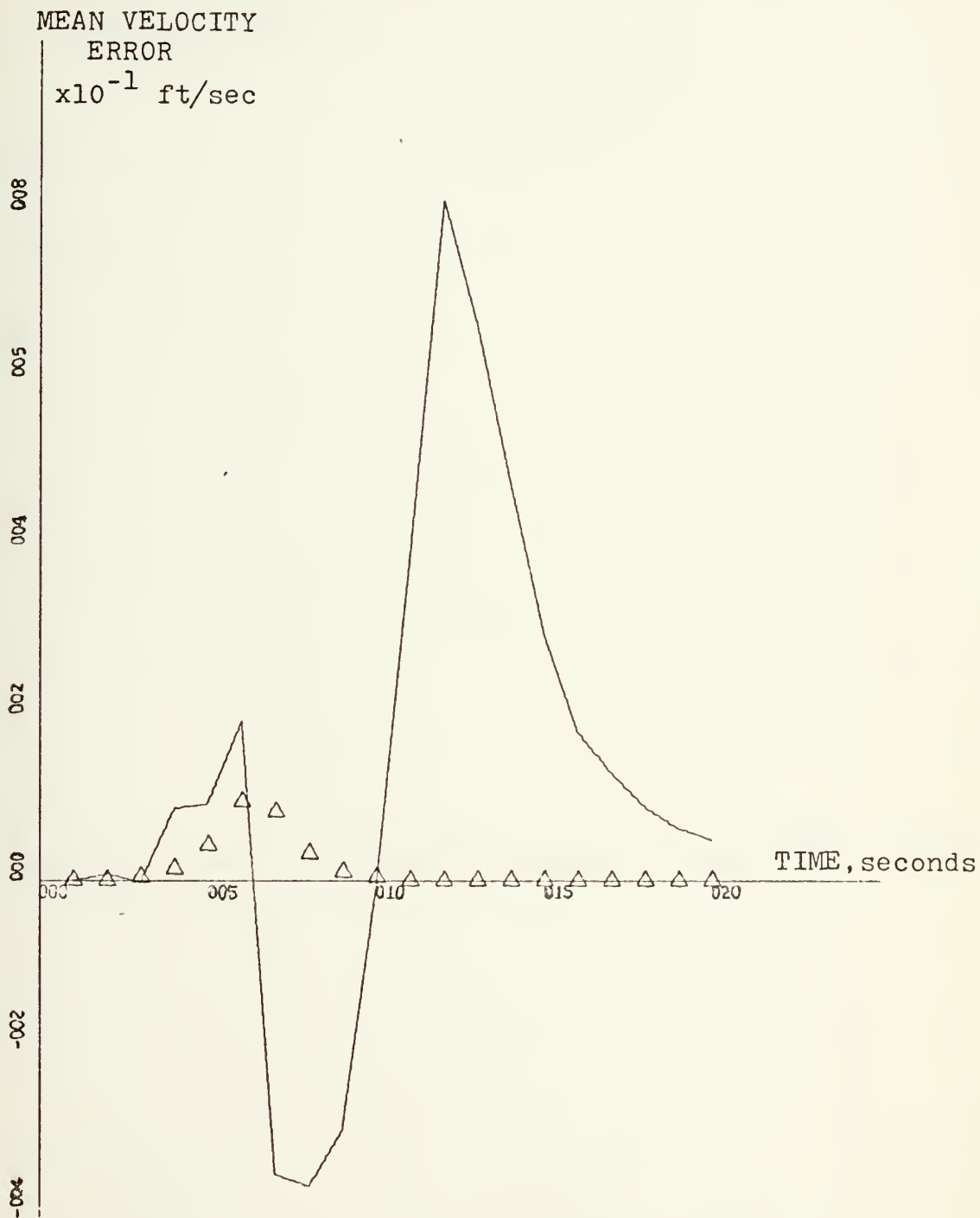


FIG. 5.12. Time history of the mean velocity estimation error for 50 Monte-Carlo runs.

X-SCALE=5.00E+00 UNITS INCH.

Y-SCALE=2.00E+01 UNITS INCH.

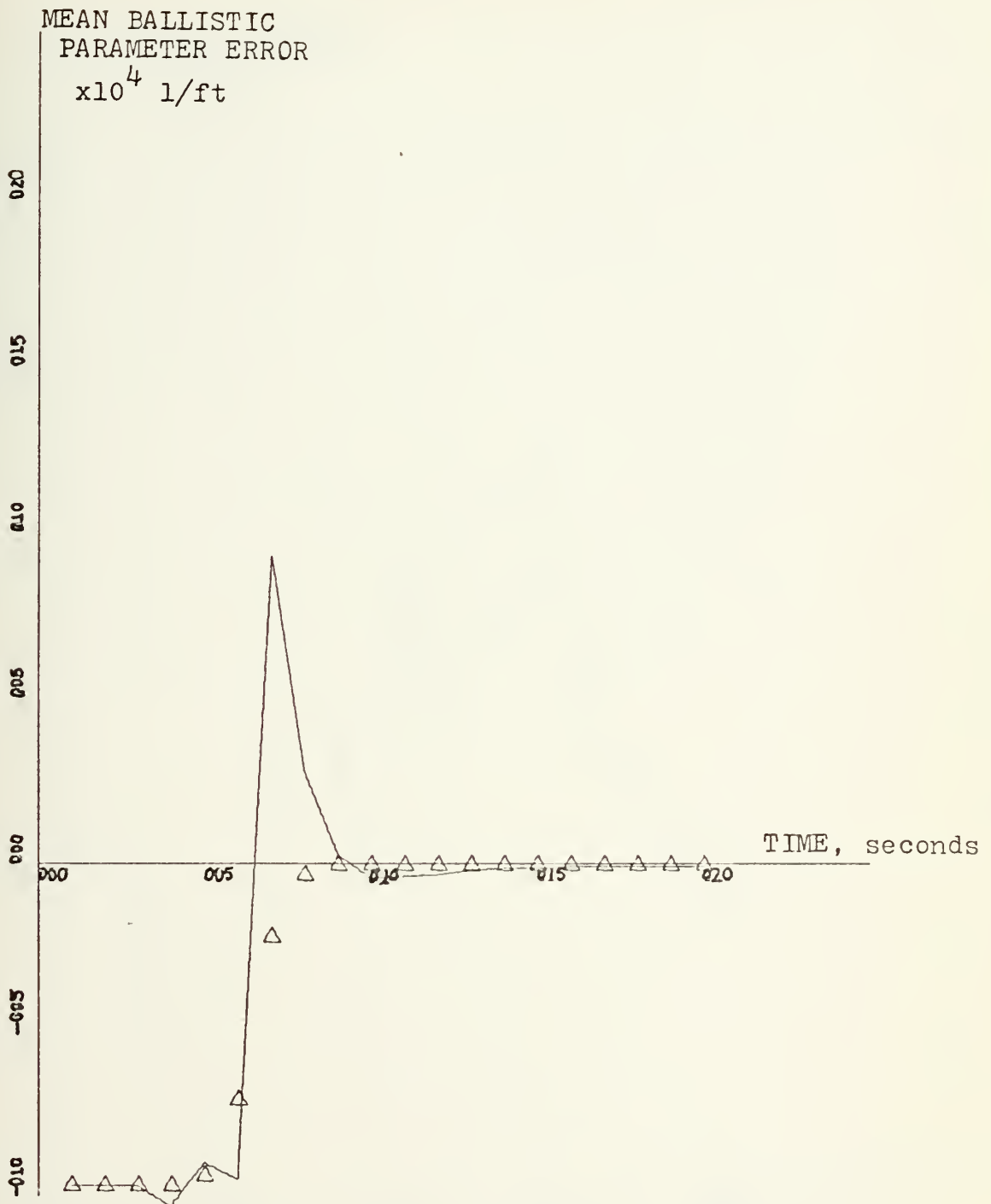


FIG. 5.13. Time history of the mean ballistic parameter estimation error for 50 Monte-Carlo runs.

X-SCALE=5.00E+00 UNITS INCH.

Y-SCALE=5.00E-04 UNITS INCH.

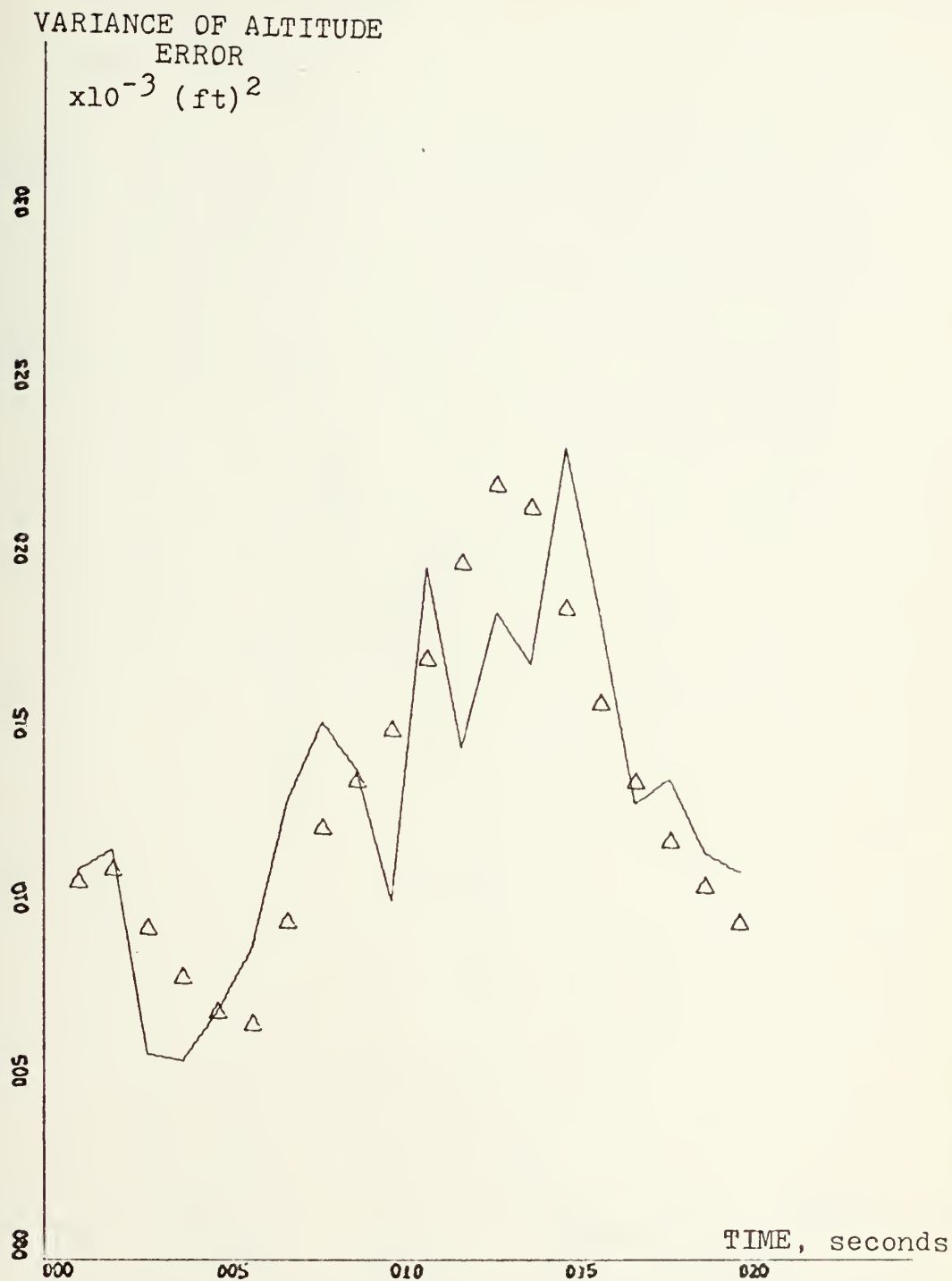


FIG. 5.14. Variance of the altitude estimation error vs. time for 50 Monte-Carlo runs.

X-SCALE=5.00E+00 UNITS INCH.

Y-SCALE=5.00E+03 UNITS INCH.

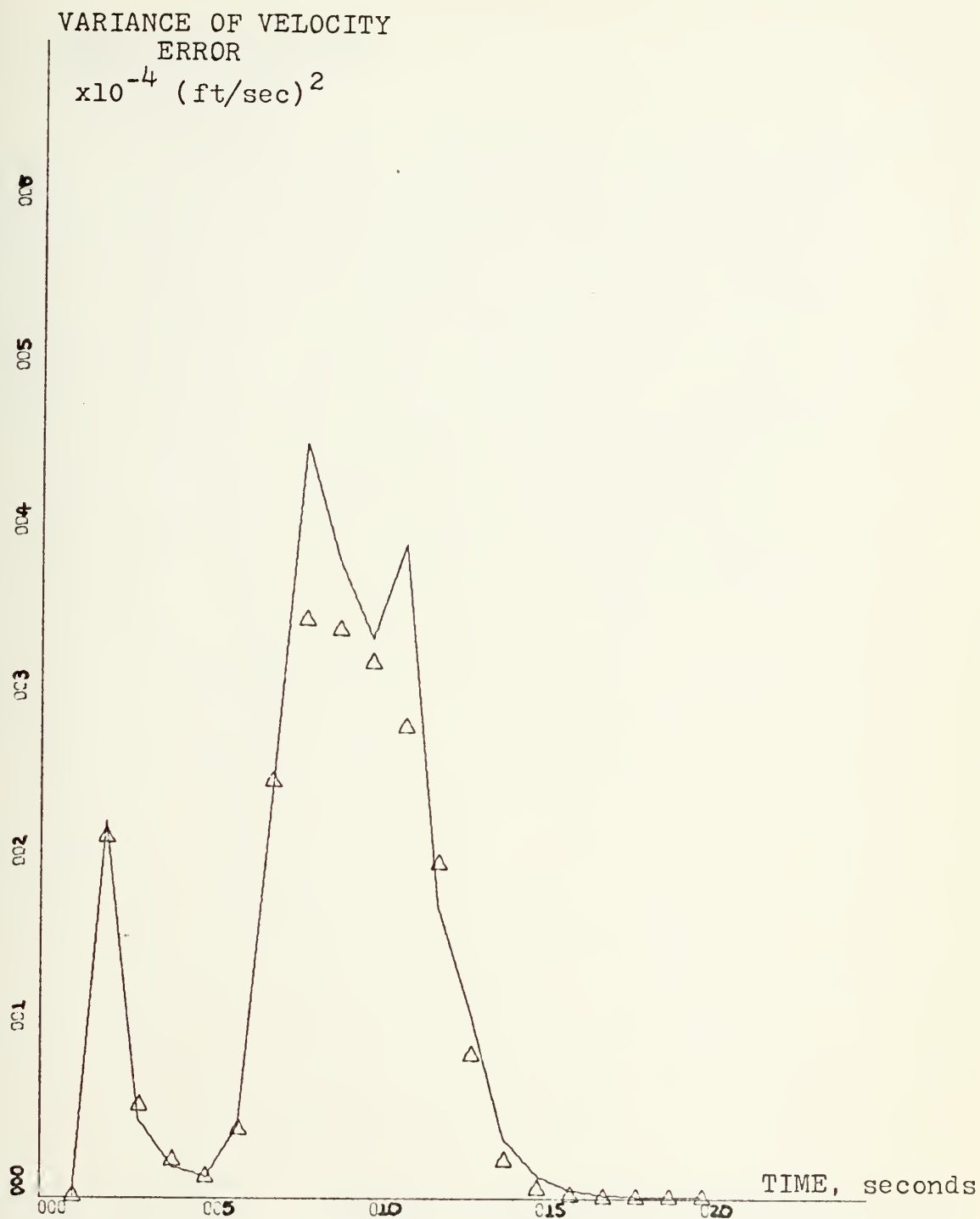


FIG. 5.15. Variance of the velocity estimation error
vs. time for 50 Monte-Carlo runs

X-SCALE=5.00E+00 UNITS INCH.

Y-SCALE=1.00E+04 UNITS INCH.

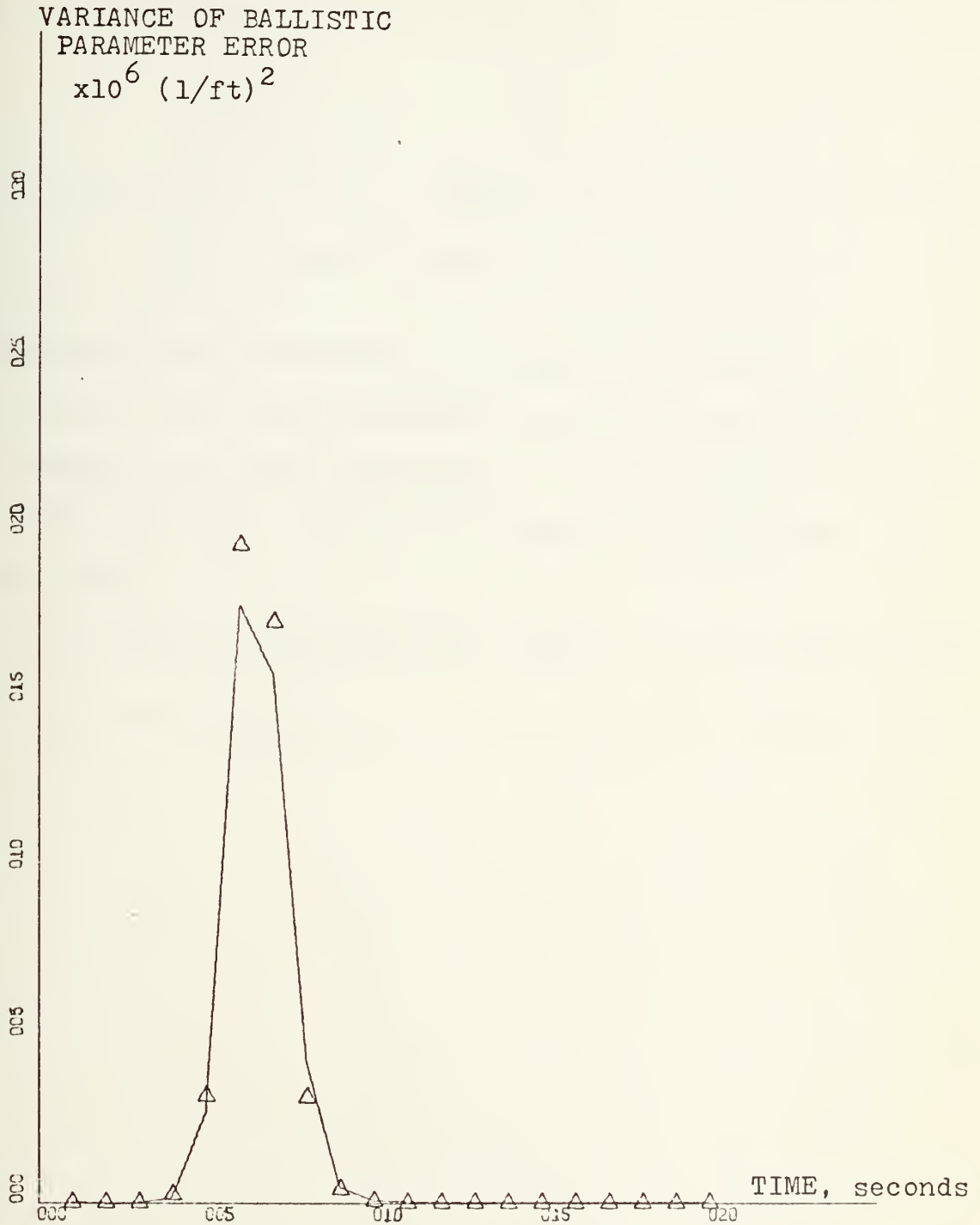


FIG. 5.16. Variance of the ballistic parameter estimation error vs. time for 50 Monte-Carlo runs.

X-SCALE:-5.00E+00 UNITS INCH.

Y-SCALE:-5.00E-06 UNITS INCH.

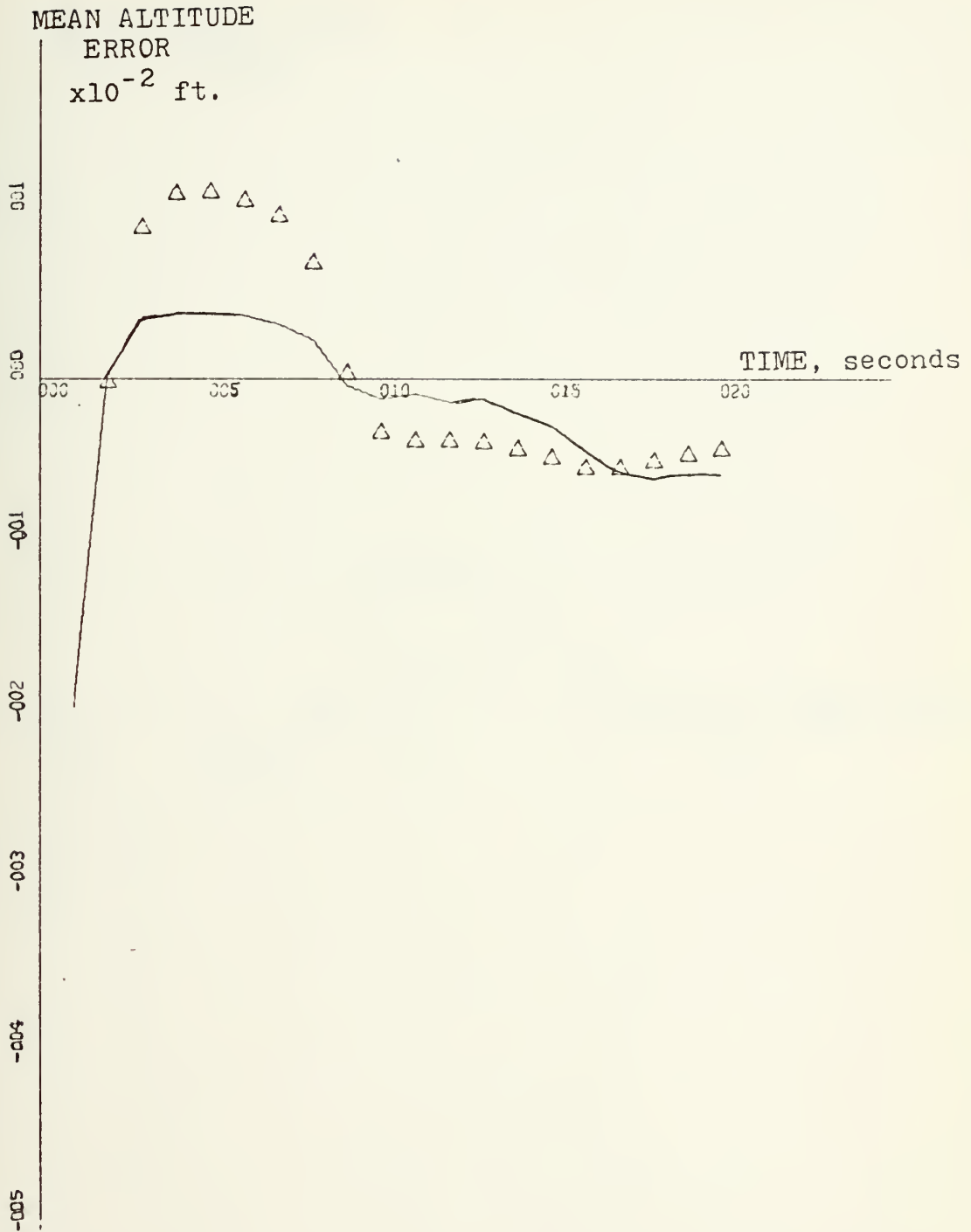
Next, the problem was solved for a different target track with initial conditions given by

$$\underline{X}(0) = \begin{bmatrix} 3.5 \times 10^5 & \text{ft.} \\ 2 \times 10^4 & \text{ft/sec} \\ 1 \times 10^{-3} & 1/\text{ft} \end{bmatrix} = \underline{b} \tag{5.35}$$

The results are illustrated in Figures 5.17 through 5.22. One can see that the analytical equations predict worse performance than that predicted by the Monte-Carlo simulation.

For comparison, the CPU times used for both method are given below.

Monte-Carlo simulation (1000 runs).....	3 min 21 sec
Analytical equations.....	0 min 9 sec



△

FIG. 5.17. Time history of altitude estimation error
for $\underline{X}(0) = \underline{b}$

X-SCALE=5.00E+00 UNITS INCH.

Y-SCALE=1.00E+02 UNITS INCH.

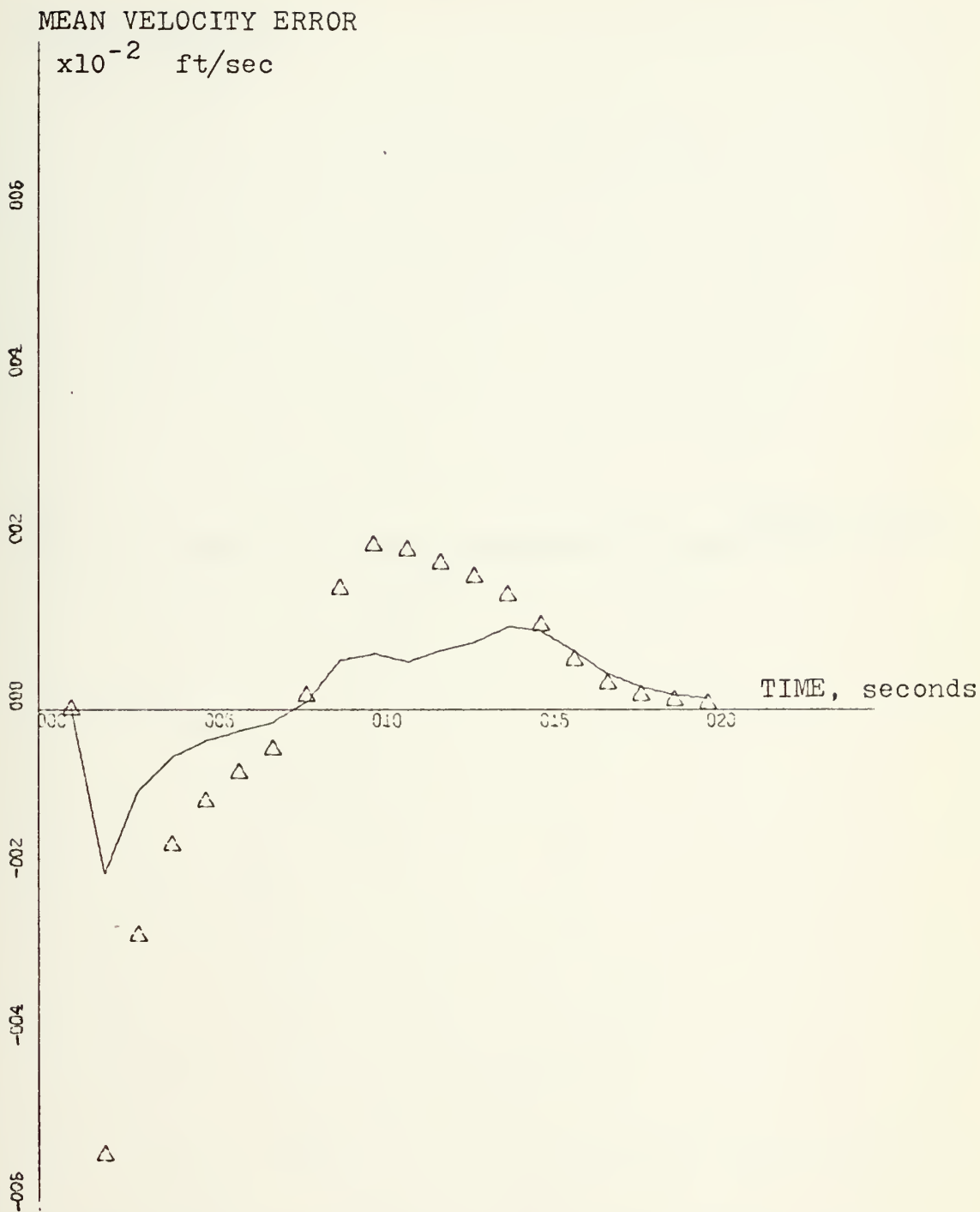


FIG. 5.18. Time history of mean velocity estimation error for $\underline{X}(0)=\underline{p}$.

X-SCALE=5.00E+00 UNITS INCH.

Y-SCALE=2.00E+02 UNITS INCH.

MEAN BALLISTIC
PARAMETER ERROR
 $\times 10^3 \text{ 1/ft.}$

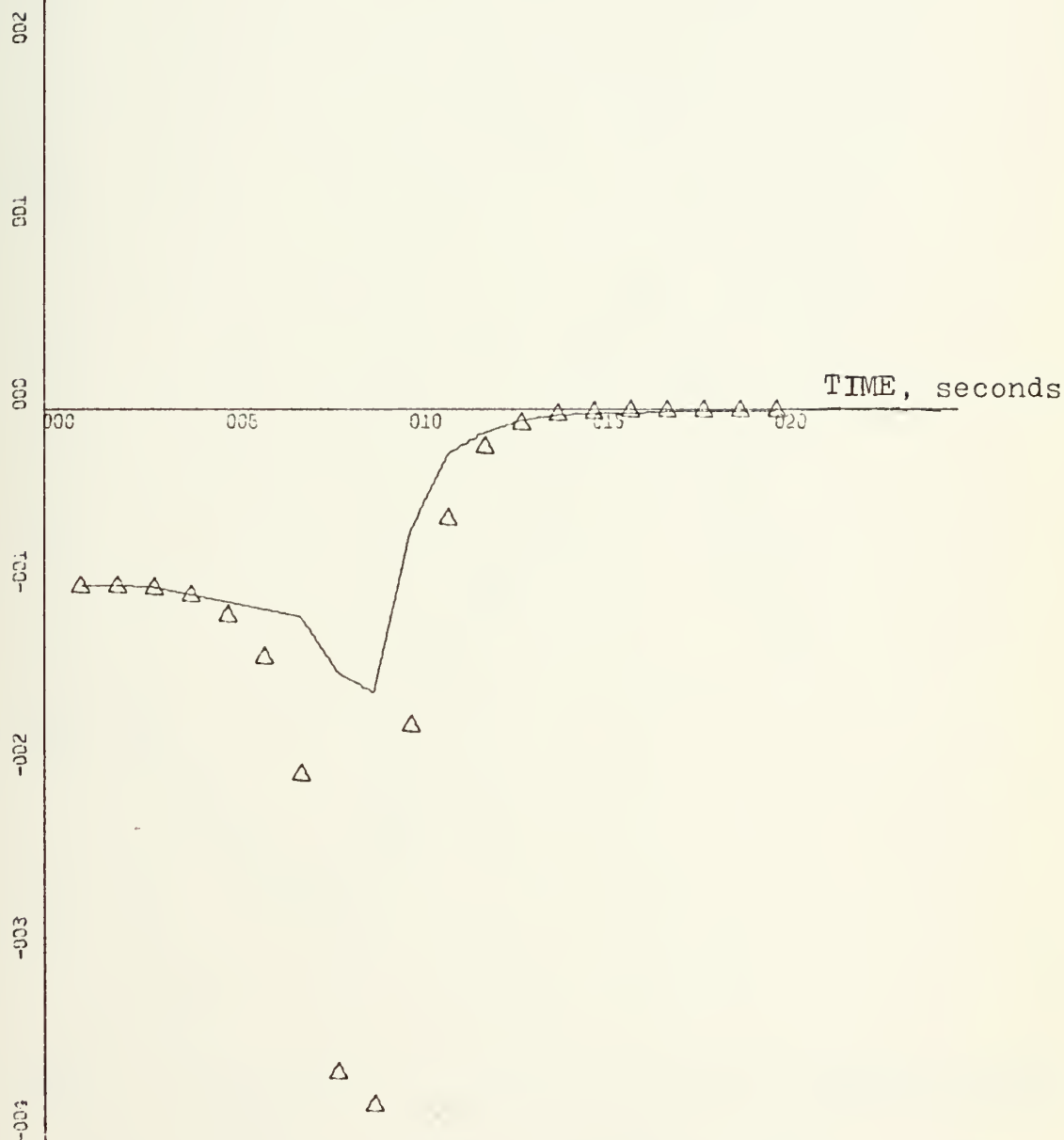


FIG. 5.19. Time history of the ballistic parameter estimation error for $\underline{X}(0) = \underline{b}$

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=1.00E-03 UNITS INCH.



FIG. 5.20. Variance of altitude estimation error
vs. time for $X(0)=b$.

X-SCALE:=5.00E+00 UNITS INCH.

Y-SCALE:=5.00E+03 UNITS INCH.

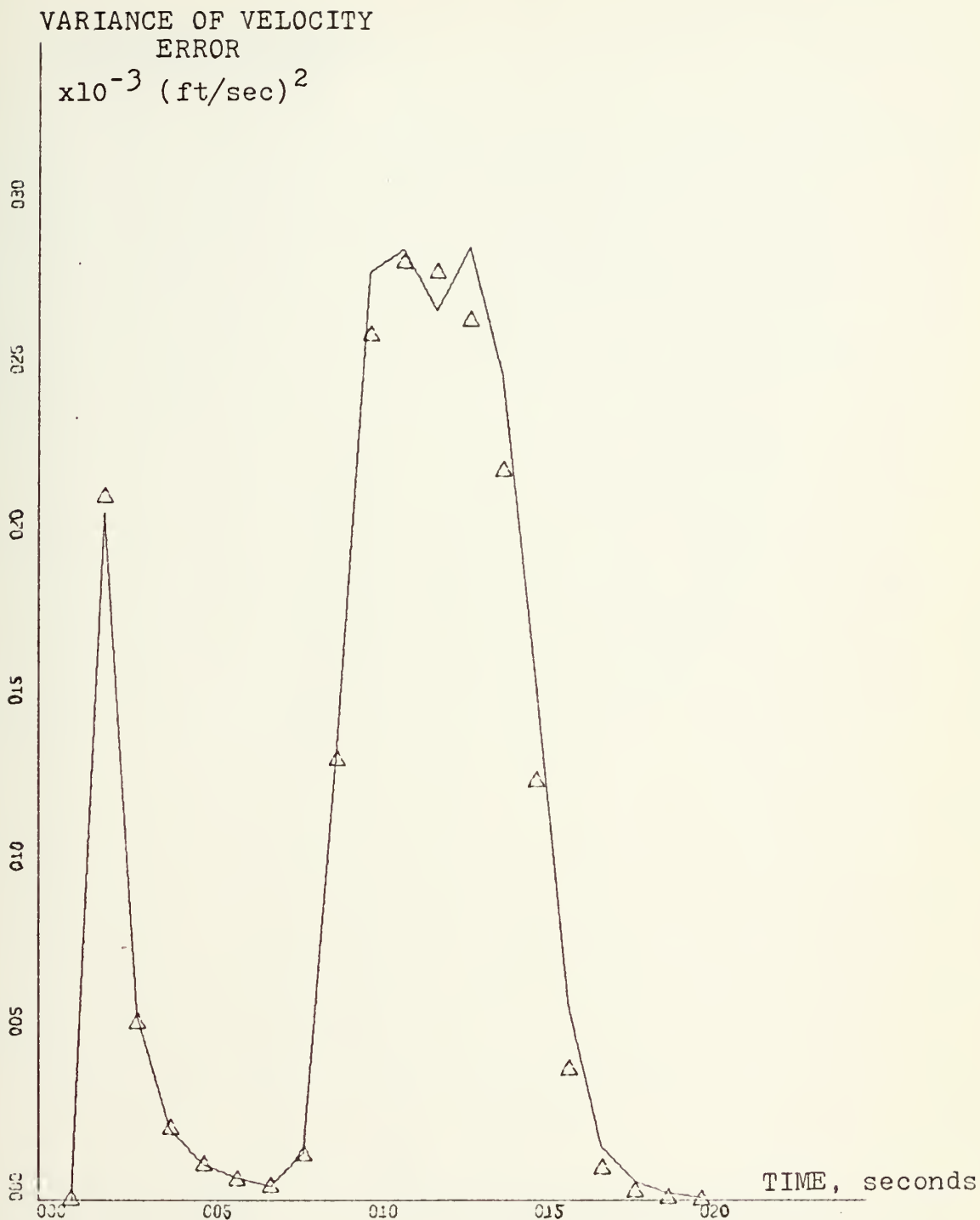


FIG. 5.21. Variance of velocity estimation error
vs. time for $X(0)=b$.

X-SCALE=5.00E+00 UNITS INCH.

Y-SCALE=5.00E+03 UNITS INCH.

VARIANCE OF BALLISTIC
PARAMETER ERROR
 $\times 10^6 \quad (1/ft)^2$

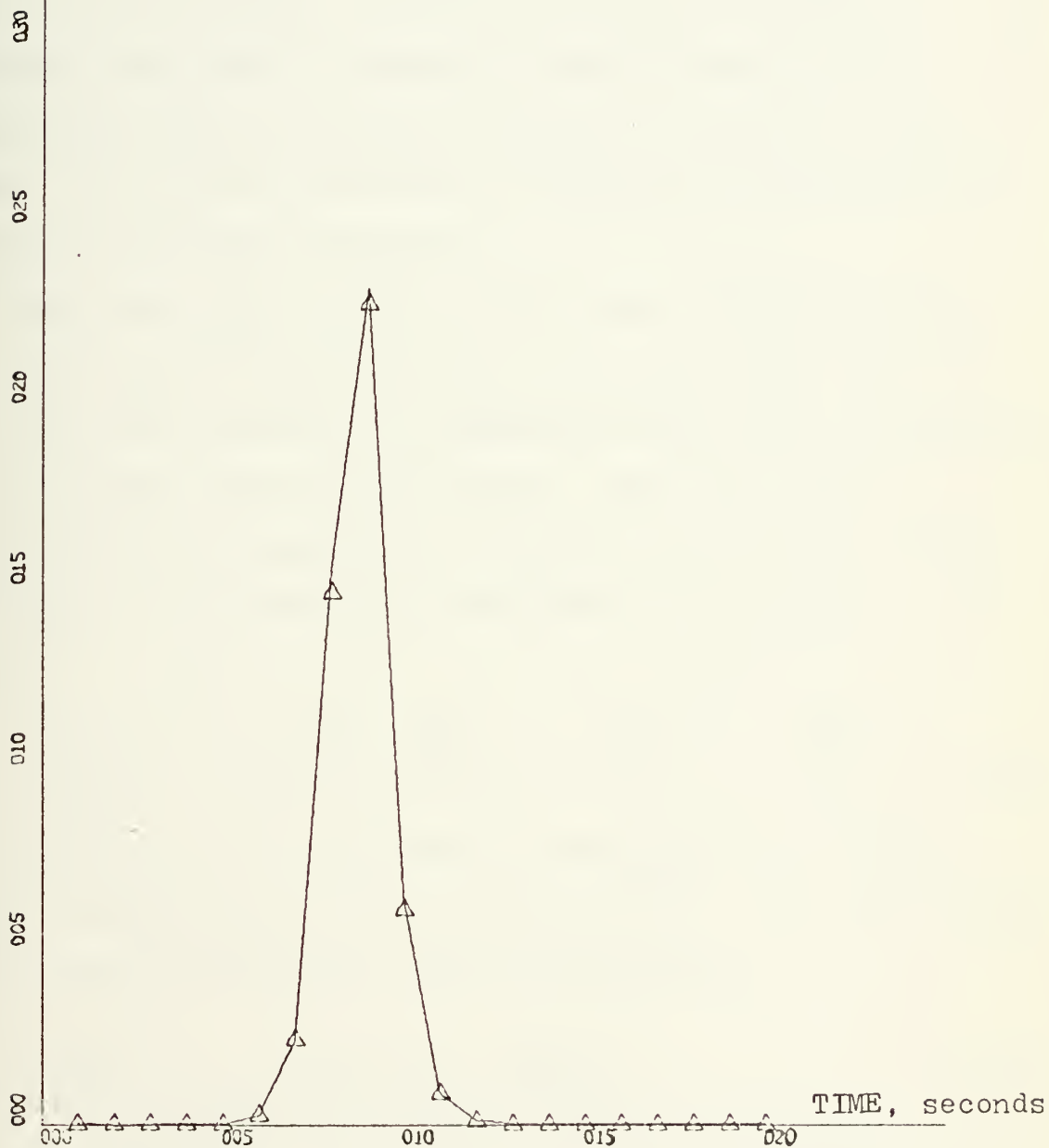


FIG. 5.22. Variance of ballistic parameter estimation error vs. time for $X(0)=b$.

X-SCALE=5.00E+00 UNITS INCH.

Y-SCALE=5.00E-06 UNITS INCH.

B. APPLICATION OF THE ANALYTICAL APPROACH TO THE MULTIPLE TRACK NONLINEAR CASE

The results derived in Chapter III can also be applied to nonlinear problems. Equation (3.11) can be used to calculate the mean of estimation error in terms of the conditional means of estimation error calculated for each track. The conditional means are computed approximately by using the analytical equations.

Since the matrices $\hat{Q}(k)$ and $\hat{H}(k)$ are calculated using the true target trajectory in the nonlinear case, one can see that the covariance of estimation error will be dependent on the tracks (see Equation (2.31)). Thus, one needs a relationship to compute the covariance of estimation error in terms of the conditional covariances of estimation error, i.e.

$$\underline{P}(k/k) = f(\overset{(1)}{\underline{P}(k/k)}, \overset{(2)}{\underline{P}(k/k)}, \dots, \overset{(i)}{\underline{P}(k/k)}, \dots, \overset{(n)}{\underline{P}(k/k)}) \quad (5.36)$$

where $\overset{(i)}{\underline{P}(k/k)}$ is the conditional covariance of estimation error corresponding to the i 'th track $\overset{(i)}{\underline{X}(k)}$. The conditional covariance of estimation error is defined as

$$\overset{(i)}{\underline{P}(k/k)} = \text{Covar} \left[\tilde{\underline{X}}(k) / \underline{X}(k) = \overset{(i)}{\underline{X}}(k) \right] \quad (5.37)$$

The relationship of the covariance to the conditional covariances is derived in the following paragraphs.

C. CONDITIONAL COVARIANCE

Given a discrete random vector \underline{B} , the expected value of the dependent continuous random vector \underline{A} can be expressed in terms of the conditional expectations as

$$E[\underline{A}] = \sum_j E[\underline{A}/\underline{B}=\underline{b}_j] \cdot P[\underline{B}=\underline{b}_j] \quad (5.38)$$

For simplicity Equation (5.38) can be written as

$$E[\underline{A}] = E \left\{ E[\underline{A}/\underline{B}] \right\}_B \quad (5.39)$$

where the subscript B denotes the expectation with respect to the random vector \underline{B} .

The conditional covariance is defined as [3]

$$\underline{P}_{A/B} = E \left\{ \left[\underline{A} - E[\underline{A}/\underline{B}] \right] \cdot \left[\underline{A} - E[\underline{A}/\underline{B}] \right]^T / \underline{B} \right\} \quad (5.40)$$

But

$$\underline{P}_A = E \left\{ \left[\underline{A} - E\{\underline{A}\} \right] \cdot \left[\underline{A} - E\{\underline{A}\} \right]^T \right\} \quad (5.41)$$

which is the (unconditional) covariance matrix. Replacing \underline{A} with

$\left[\underline{A} - E\{\underline{A}\} \right] \cdot \left[\underline{A} - E\{\underline{A}\} \right]^T$ in Equation (5.39) gives **

$$\underline{P}_A = E \left\{ E \left\{ \left[\underline{A} - E\{\underline{A}\} \right] \cdot \left[\underline{A} - E\{\underline{A}\} \right]^T / \underline{B} \right\} \right\}_B \quad (5.42)$$

**Equation (5.39) can be written in the general case as

$E\{g(a)\} = E \left\{ E[g(a)/\underline{A}] \right\}$ where $g(a)$ may be any function.

By adding and subtracting $E \left[\underline{A}/\underline{B} \right]$ inside the parantheses,
Equation (5.42) can be written as

$$P_A = E \left\{ E \left\{ \left[\left(\underline{A} - E \left[\underline{A}/\underline{B} \right] \right) + \left(E \left[\underline{A}/\underline{B} \right] - E \left[\underline{A} \right] \right) \right] \cdot \right. \right. \\ \left. \left. \left[\left(\underline{A} - E \left[\underline{A}/\underline{B} \right] \right) + \left(E \left[\underline{A}/\underline{B} \right] - E \left[\underline{A} \right] \right) \right]^T / \underline{B} \right\} \right\}_B$$

or

$$P_A = E \left\{ E \left\{ \left[\begin{aligned} &(\underline{A} - E \left[\underline{A}/\underline{B} \right]) \cdot (\underline{A} - E \left[\underline{A}/\underline{B} \right])^T \\ &+ (\underline{A} - E \left[\underline{A}/\underline{B} \right]) \cdot (E \left[\underline{A}/\underline{B} \right] - E \left[\underline{A} \right])^T + (E \left[\underline{A}/\underline{B} \right] - E \left[\underline{A} \right]) \\ &\cdot (\underline{A} - E \left[\underline{A}/\underline{B} \right])^T + (E \left[\underline{A}/\underline{B} \right] - E \left[\underline{A} \right]) \\ &\cdot (E \left[\underline{A}/\underline{B} \right] - E \left[\underline{A} \right])^T \end{aligned} \right] / \underline{B} \right\} \right\}_B \quad (5.43)$$

Using the properties of the expectation operator in the
inner expectation, the cross terms can be calculated as

$$E \left\{ (\underline{A} - E \left[\underline{A}/\underline{B} \right]) \cdot (E \left[\underline{A}/\underline{B} \right] - E \left[\underline{A} \right])^T / \underline{B} \right\} = E \left\{ (\underline{A} - E \left[\underline{A}/\underline{B} \right]) / \underline{B} \right\} \\ \cdot (E \left[\underline{A}/\underline{B} \right] - E \left[\underline{A} \right])^T \quad (5.44)$$

(because $E \left[\underline{A} \right]$ and $E \left[\underline{A}/\underline{B} \right]$ are not functions of \underline{A} , they are
deterministic with respect to \underline{A} for a given value of \underline{B}).

This means $E_A \left\{ E \left[\underline{A}/\underline{B} \right] \right\} = E \left[\underline{A} \right]$ and $E_A \left\{ E \left[\underline{A}/\underline{B} \right] / \underline{B} \right\} = E_A \left\{ \underline{A}/\underline{B} \right\}$.

Using the properties of the expectation operator yields

$$\begin{aligned}
 E \left\{ (\underline{A} - E[\underline{A}/\underline{B}]) / \underline{B} \right\} &= E[\underline{A}/\underline{B}] - E \left\{ E[\underline{A}/\underline{B}] / \underline{B} \right\} \\
 &= (E[\underline{A}/\underline{B}] - E[\underline{A}/\underline{B}]) \\
 &= \underline{0}
 \end{aligned} \tag{5.45}$$

Thus, the cross terms drop out in Equation (5.43). Since $(E[\underline{A}/\underline{B}] - E[\underline{A}])$ is not a function of \underline{A} but it is a function of \underline{B} , then Equation (5.43) reduces to

$$\begin{aligned}
 \underline{P}_A = E \left\{ E \left\{ (\underline{A} - E[\underline{A}/\underline{B}]) \cdot (\underline{A} - E[\underline{A}/\underline{B}])^T / \underline{B} \right\} \right. \\
 \left. + (E[\underline{A}/\underline{B}] - E[\underline{A}]) \cdot (E[\underline{A}/\underline{B}] - E[\underline{A}])^T \right\}_B
 \end{aligned} \tag{5.46}$$

or

$$\begin{aligned}
 \underline{P}_A = E \left\{ E \left\{ (\underline{A} - E[\underline{A}/\underline{B}]) \cdot (\underline{A} - E[\underline{A}/\underline{B}])^T / \underline{B} \right\} \right\}_B \\
 + E \left\{ (E[\underline{A}/\underline{B}] - E[\underline{A}]) \cdot (E[\underline{A}/\underline{B}] - E[\underline{A}])^T \right\}_B
 \end{aligned} \tag{5.47}$$

The inner expectation in the first term of the right hand side defines the conditional covariance (see Equation (5.40), hence

$$\underline{P}_A = E \left\{ \underline{P}_{A/B} \right\}_B + E \left\{ (E[\underline{A}/\underline{B}] - E[\underline{A}]) \cdot (E[\underline{A}/\underline{B}] - E[\underline{A}])^T \right\}_B \tag{5.48}$$

Equation (5.48) gives the required relationship between the covariance and conditional covariances, i.e.

$$\begin{aligned}
\underline{P}(k/k) &= E \left\{ \text{Covar} \left\{ \tilde{\underline{X}}(k) / \underline{X}(k) = \underline{X}^{(i)}(k) \right\} \right\}_{\underline{X}(k)} \\
&\quad + E \left\{ \left(\underline{\mu}^{(i)}(k) - \underline{\mu}(k) \right) \cdot \left(\underline{\mu}^{(i)}(k) - \underline{\mu}(k) \right)^T \right\}_{\underline{X}(k)} \\
&= E \left\{ \underline{P}(k/k) \right\}_{\underline{X}(k)} + E \left\{ \left(\underline{\mu}^{(i)}(k) - \underline{\mu}(k) \right) \right. \\
&\quad \left. \cdot \left(\underline{\mu}^{(i)}(k) - \underline{\mu}(k) \right)^T \right\}_{\underline{X}(k)}
\end{aligned}$$

which can also be expressed as

$$\begin{aligned}
\underline{P}(k/k) &= \sum_{i=1}^n \underline{P}(k/k)^{(i)} \cdot p_i + \sum_{i=1}^n \left[\underline{\mu}^{(i)}(k) - \tilde{\underline{\mu}}(k) \right] \\
&\quad \cdot \left[\tilde{\underline{\mu}}(k) - \underline{\mu}(k) \right]^T \cdot p_i \quad (5.49)
\end{aligned}$$

where

$\tilde{\underline{\mu}}^{(i)}(k)$ and $\underline{P}(k/k)^{(i)}$ are the conditional mean and covariance of estimation error for the i 'th track $\underline{X}^{(i)}(k)$,

$\tilde{\underline{\mu}}(k)$ and $\underline{P}(k/k)$ are the overall mean and covariance of estimation error and

p_i is the probability of occurrence of the i 'th track $\underline{X}^{(i)}(k)$.

First, using Equation (3.11) one can compute the mean of estimation error $\underline{\mu}(k)$. Then using Equation (5.49) the covariance can be calculated.

The re-entry problem was simulated for two tracks

where

$$\begin{aligned} p_1 &= P \{ \underline{X}^{(1)}(k) = \underline{X}^{(1)}(k) \} \\ &= 0.5 \end{aligned} \quad (5.50)$$

$$\begin{aligned} p_2 &= P \{ \underline{X}^{(2)}(k) = \underline{X}^{(2)}(k) \} \\ &= 0.5 \end{aligned} \quad (5.51)$$

with

$$\underline{X}^{(1)}(0) = \begin{bmatrix} 3 \times 10^5 & \text{ft.} \\ 2 \times 10^4 & \text{ft/sec} \\ 1 \times 10^{-3} & 1/\text{ft} \end{bmatrix} \quad (5.52)$$

$$\underline{X}^{(2)}(0) = \begin{bmatrix} 3.5 \times 10^5 & \text{ft.} \\ 2 \times 10^4 & \text{ft/sec} \\ 1 \times 10^{-3} & 1/\text{ft} \end{bmatrix} \quad (5.53)$$

Figures 5.23 through 5.28 illustrate the results. The continuous curves represent the Monte-Carlo simulation and the triangles represent the results from the analytical equations. From the figures one can see that the 1000-member ensemble Monte-Carlo simulation predicts better performance. Actually the differences between the two results are small compared to the values of the state variables. For example, the maximum difference between the two results is 150 feet for

the altitude error at 300,000 feet altitude (which is the initial difference; at later times the two results are closer). The CPU times used for these simulations are given below.

	<u>CPU Time</u>
Monte-Carlo simulation (1000 runs).....	3 min 50 sec
Monte-Carlo simulation (50 runs).....	0 min 17 sec
Analytical equations.....	0 min 7 sec

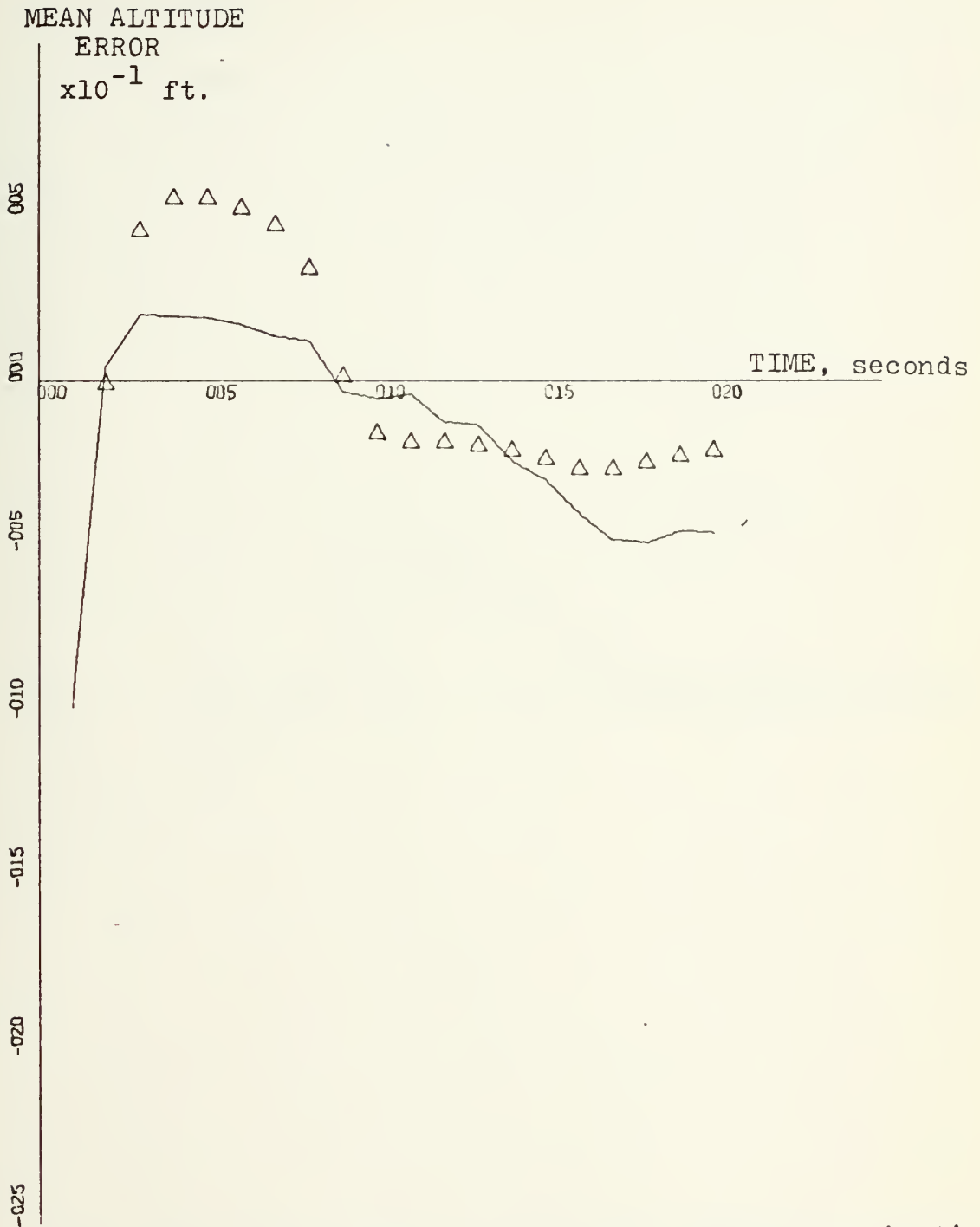


FIG. 5.23. Time history of the mean altitude estimation error for two tracks.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=5.00E+01 UNITS INCH.

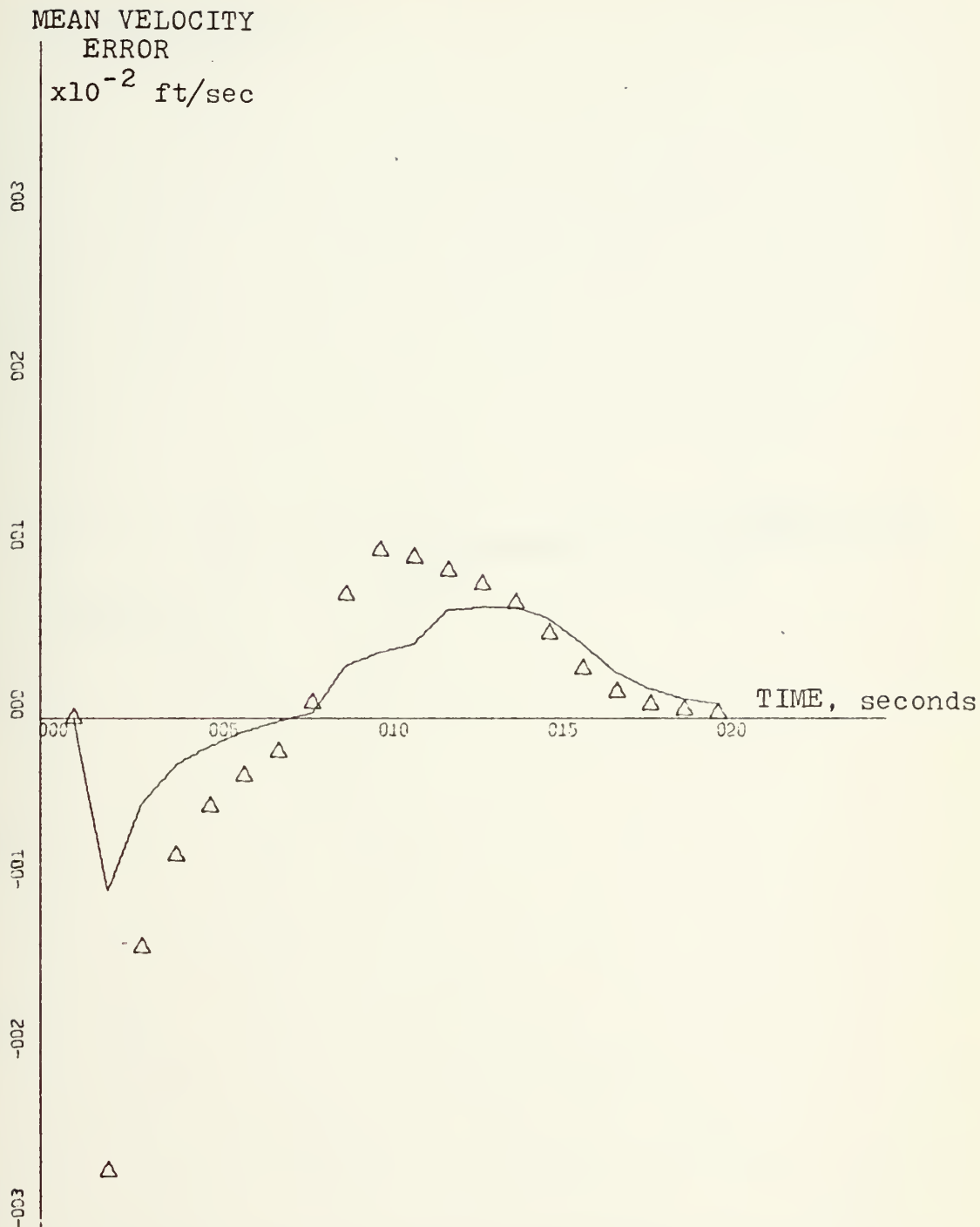


FIG. 5.24. Time history of the mean velocity estimation error for two tracks.

X-SCALE=5.00E+00 UNITS INCH.

Y-SCALE=1.00E+02 UNITS INCH.

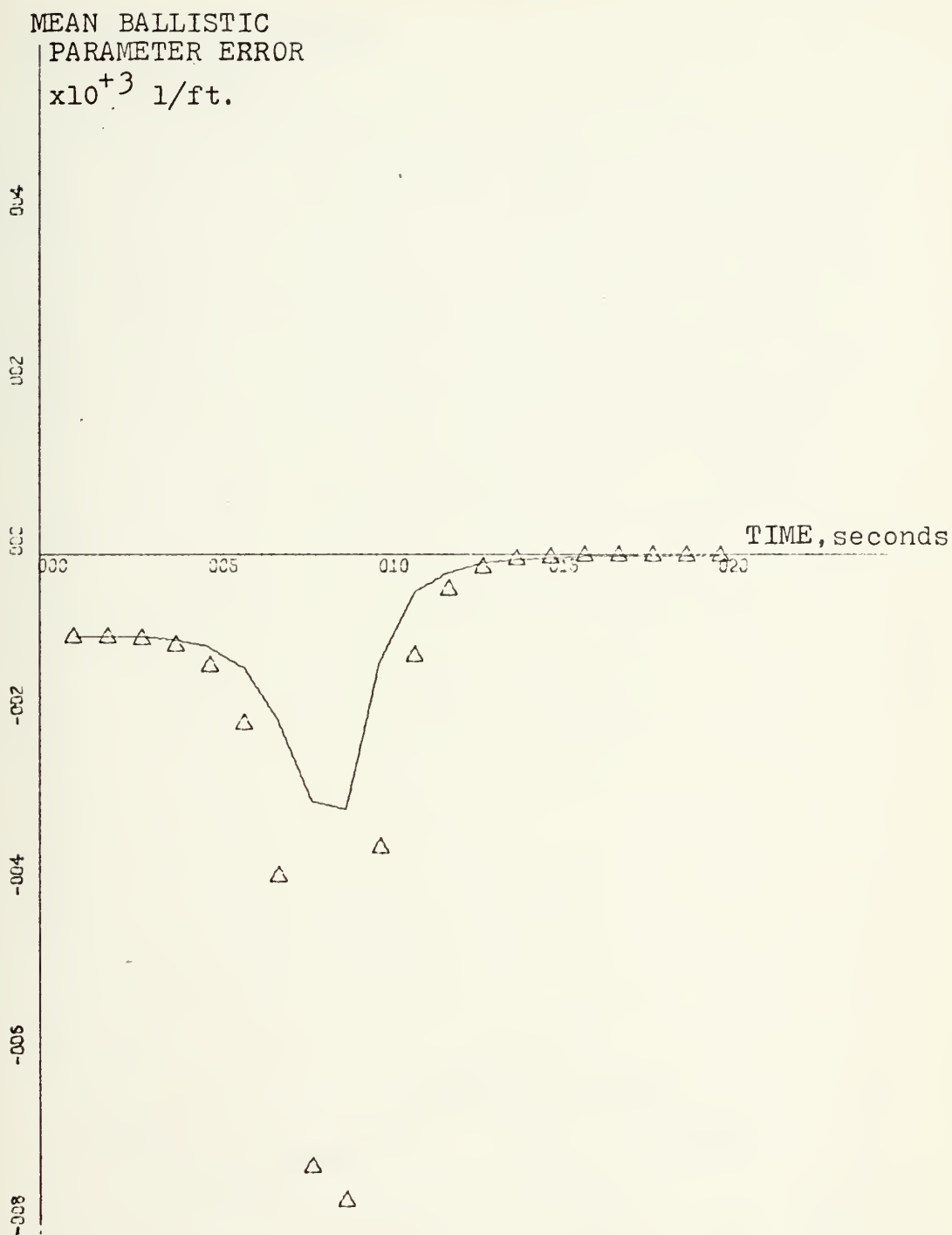


FIG. 5.25. Time history of the mean ballistic parameter estimation error for two tracks.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=2.00E-03 UNITS INCH.



FIG. 5.26. Time history of the variance of the altitude estimation error for two tracks.

X-SCALE=5.00E+00 UNITS INCH.

Y-SCALE=2.00E+04 UNITS INCH.

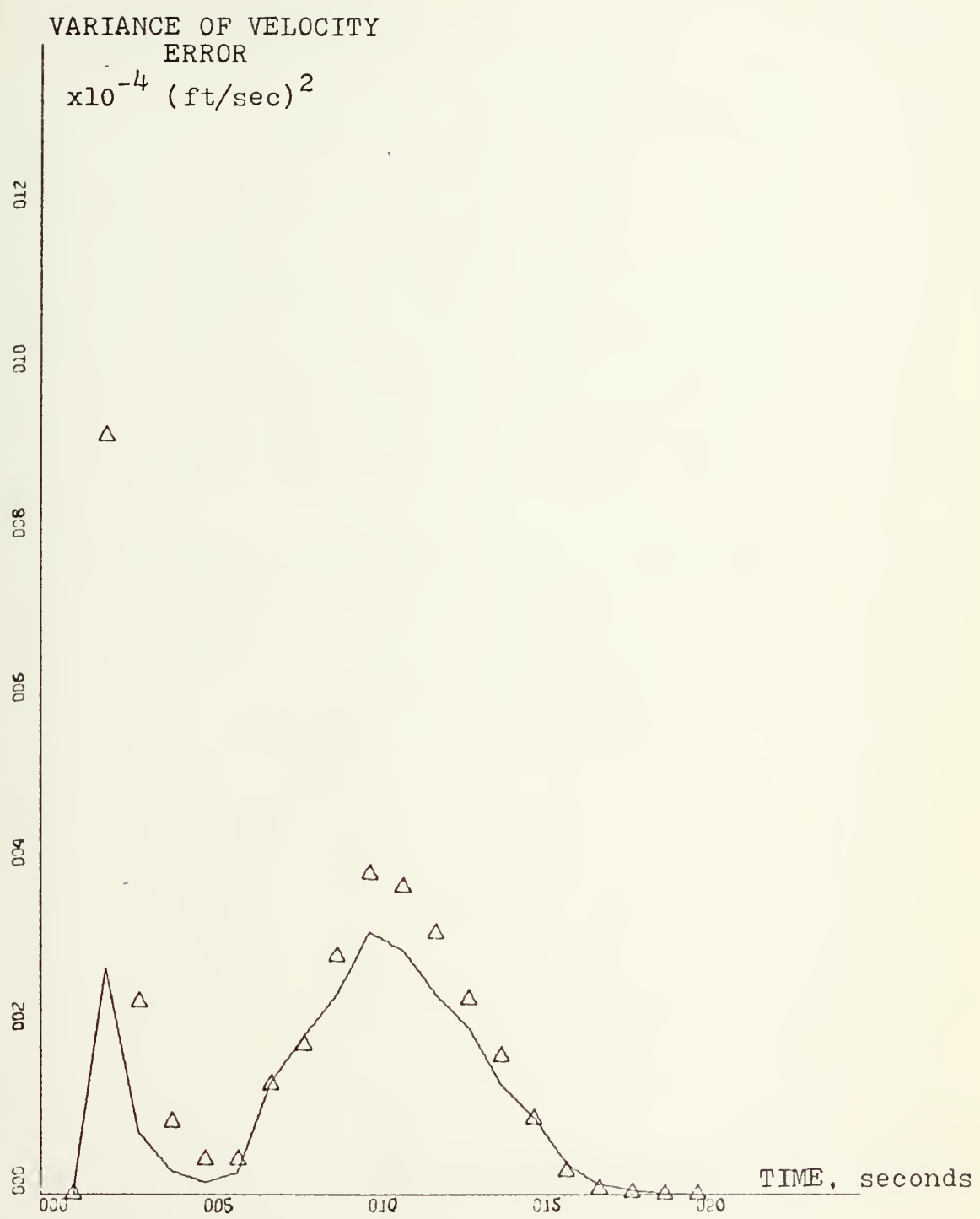


FIG. 5.27. Time history of the variance of the velocity estimation error for two tracks.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=2.00E+04 UNITS INCH.

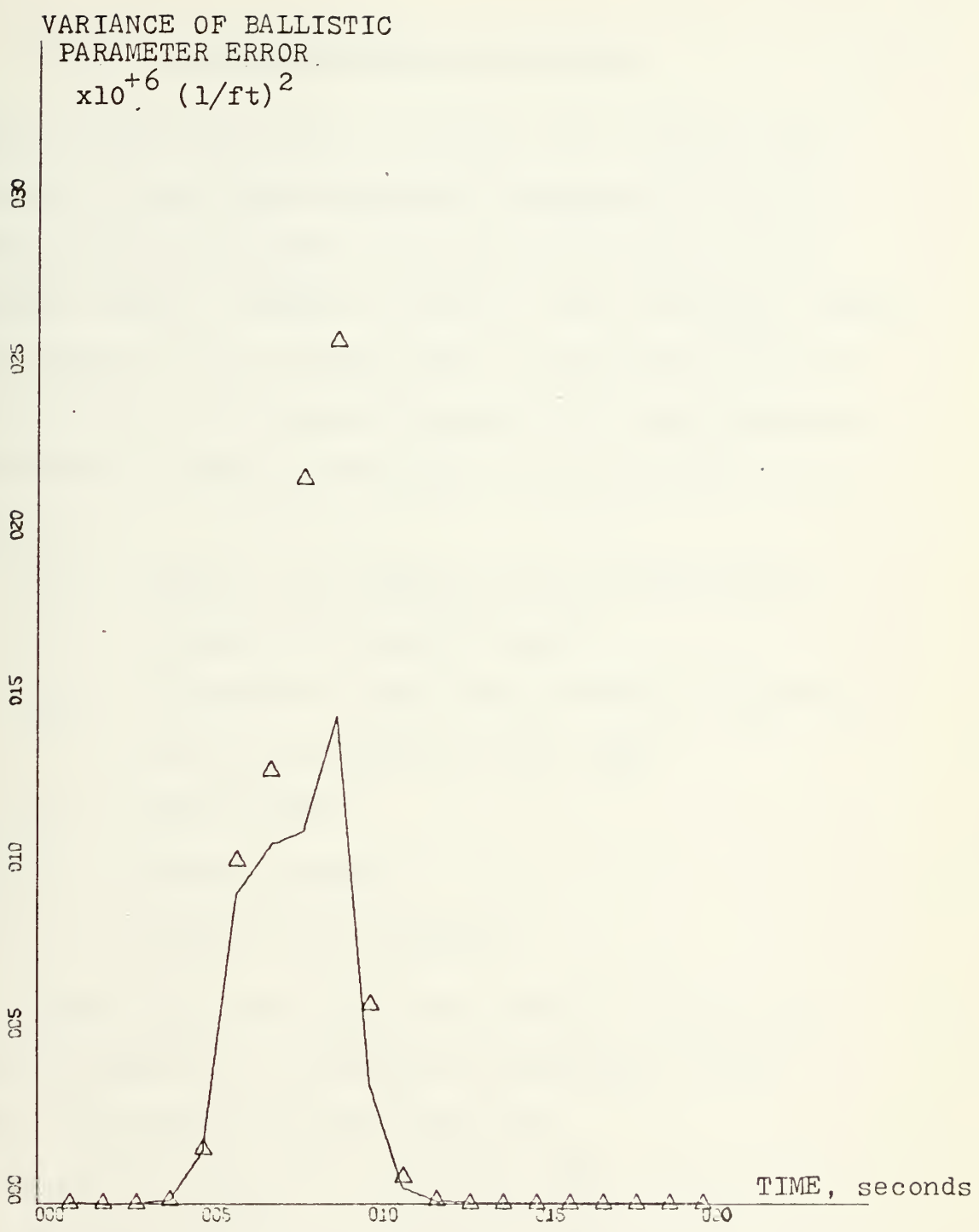


FIG. 5.28. Time history of the variance of the ballistic parameter estimation error for two tracks.
X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=5.00E-06 UNITS INCH.

VI. A SUBMARINE TRACKING PROBLEM

This chapter discusses an example with linear state equations but nonlinear measurement equations. It is a problem in which estimates are made of the states of a submerged target: heading, velocity, rest frequency, range to the target at the closest point of approach (cpa) and distance to cpa. One passive sensor (sono bouy) provides measurements of target heading and frequency [4]. The states are

- R_{cpa} : Range to the target at the closest point of approach from the sensor.
- X_{cpa} : X distance to cpa from the sensor (negative before cpa, positive after cpa).
- V_s : Target speed.
- θ_s : Target heading.
- F_o : Target rest frequency.

It is assumed that the target has constant velocity and heading. Figure 6.1 illustrates the geometry of the problem. The state equations are [4]

$$\begin{aligned}
 R_{cpa}(k+1) &= R_{cpa}(k) + g_1 (\dot{\gamma}_{V_s}, \dot{\gamma}_{\theta_s}, k) \\
 X_{cpa}(k+1) &= X_{cpa}(k) + V_s \cdot T + g_2 (\dot{\gamma}_{V_s}, \dot{\gamma}_{\theta_s}, k) \\
 V_s(k+1) &= V_s(k) + g_3 (\dot{\gamma}_{V_s})
 \end{aligned} \tag{6.1}$$

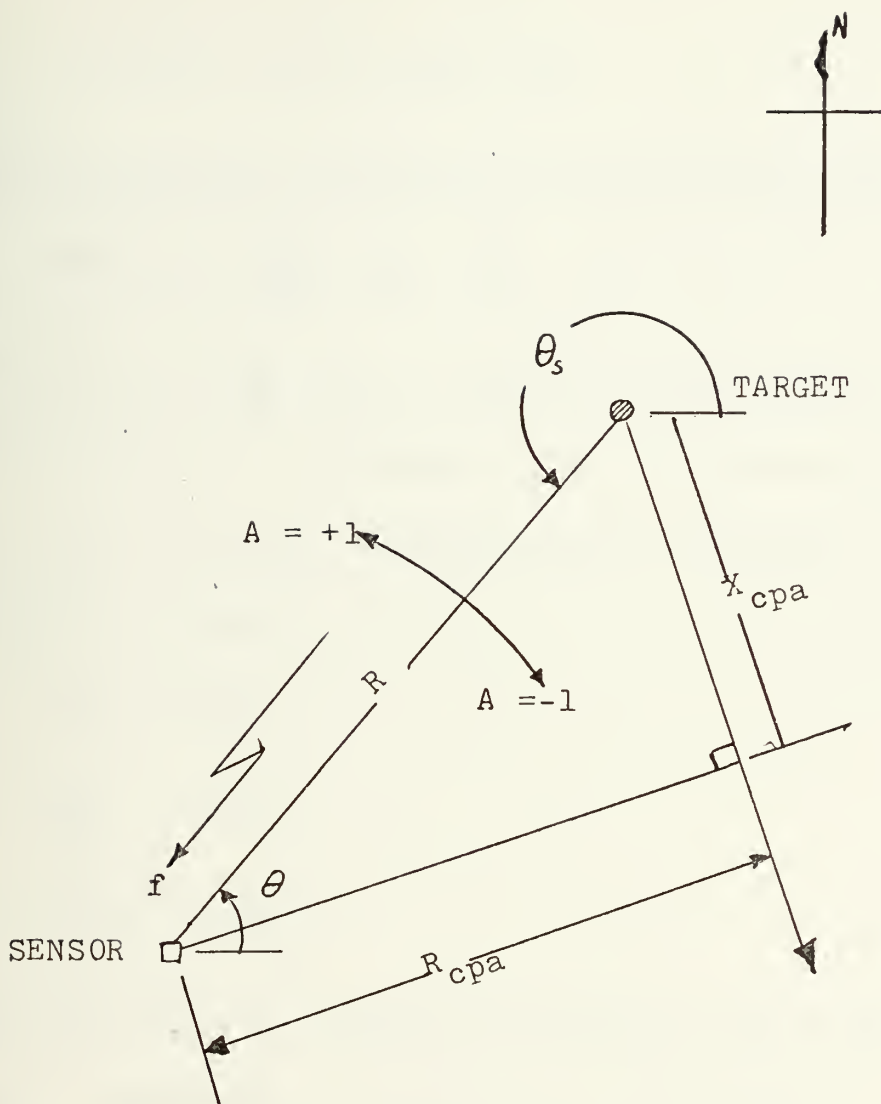


FIG. 6.1. $R_{cpa} - X_{cpa}$ filter geometry.

$$\underline{s}^{(k+1)} = \underline{\theta}_s(k) + g_4(\dot{\gamma}_{\underline{\theta}_s})$$

$$F_o(k+1) = F_o(k) + g_5(\dot{\gamma}_{F_o})$$

with g_1 through g_5 the random forcing terms, hence,

$$\underline{W}(k) = g(\dot{\gamma}_{\underline{\theta}_s}, \dot{\gamma}_{V_s}, \dot{\gamma}_{F_o}, k) \quad (6.2)$$

The quantities $\dot{\gamma}_{\underline{\theta}_s}$, $\dot{\gamma}_{V_s}$, $\dot{\gamma}_{F_o}$ are random changes in heading, velocity and rest frequency. They are assumed to be independent, zero mean, piecewise-constant rates of change. They have variances defined by

$$\Sigma_{V_s}^2 = E\{\dot{\gamma}_{V_s}^2\}$$

$$\Sigma_{\underline{\theta}_s}^2 = E\{\dot{\gamma}_{\underline{\theta}_s}^2\}$$

$$\Sigma_{F_o}^2 = E\{\dot{\gamma}_{F_o}^2\}$$

The values for the standard deviations were taken from typical maneuvering parameters for the target

$$\Sigma_{\underline{\theta}_s} = 100^\circ / 1000 \text{ sec} = 1.74533 \times 10^{-3} \text{ rad/sec}$$

$$\Sigma_{V_s} = 10 \text{ kts} / 1000 \text{ sec} = 5.5555 \times 10^{-3} \text{ yds/sec}^2 \quad (6.3)$$

$$\Sigma_{F_o} = 0.5 \text{ Hz} / 1000 \text{ sec} = 0.5 \times 10^{-3} \text{ Hz/sec}$$

With the expressions for the random forcing terms included, the state equation become [4]

$$R_{cpa}(k+1) = R_{cpa}(k) + A \cdot X_{cpa}(k) \cdot \dot{\gamma}_{\theta_s} \cdot T$$

$$X_{cpa}(k+1) = X_{cpa}(k) + V_s(k) \cdot T + \frac{1}{2} \cdot \dot{\gamma}_{V_s}$$

$$\cdot T^2 - A \cdot R_{cpa}(k) \cdot \dot{\gamma}_{\theta_s} \cdot T$$

$$V_s(k+1) = V_s(k) + \dot{\gamma}_{V_s} \cdot T$$

(6.4)

$$\theta_s(k+1) = \theta_s(k) + \dot{\gamma}_{\theta_s} \cdot T$$

$$F_o(k+1) = F_o(k) + \dot{\gamma}_{F_o} \cdot T$$

where A is +1 for counter-clockwise rotation about the sensor and -1 for clockwise rotation about the sensor. A is needed to give the correct sign for a given geometry.

The angle measurement equation is

$$\begin{aligned} \theta(k) &= \theta_s(k) - \tan^{-1} \left[\frac{A \cdot R_{cpa}(k)}{X_{cpa}(k)} \right] + \theta(k) \quad \text{for } X_{cpa} < 0 \\ &= \theta_s(k) - \tan^{-1} \left[\frac{A \cdot R_{cpa}(k)}{X_{cpa}(k)} \right] - 180^\circ + \theta(k) \quad \text{for } X_{cpa} > 0 \end{aligned} \quad (6.5)$$

The frequency observation equation is

$$f(k) = \frac{F_o(k) \cdot V_p}{V_p + \frac{V_s(k) \cdot X_{cpa}(k)}{\sqrt{R_{cpa}(k)^2 + X_{cpa}(k)^2}}} + V_f(k) \quad (6.6)$$

where $V_{\theta}(k)$, $V_f(k)$ are measurement noises and V_p is the velocity of sound in the medium (V_p is assumed to be 1640 yds/sec). The excitation matrix $\underline{Q}(k)$ can be found as [4]

$$\underline{Q}(k) = E\{\underline{W}(k) \cdot \underline{W}(k)^T\}$$

$$\underline{Q}(k) = \begin{bmatrix} X_{cpa} \cdot T^2 \cdot \Sigma_{\dot{\theta}_s}^2 & -R_{cpa} \cdot X_{cpa} \cdot T^2 \cdot \Sigma_{\dot{\theta}_s}^2 & 0 & A \cdot X_{cpa} \cdot T^2 \cdot \Sigma_{\dot{\theta}_s}^2 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \left[\begin{array}{cc|cc} \frac{T^4}{4} \Sigma_{\dot{V}_s}^2 + R_{cpa}^2 & T^2 \cdot \Sigma_{\dot{\theta}_s}^2 & -\frac{T^3}{2} \Sigma_{V_s}^2 & -A \cdot R_{cpa} \cdot T^2 \cdot \Sigma_{\dot{\theta}_s}^2 & 0 \\ \hline \text{SYMMETRIC} & & T^2 \cdot \Sigma_{\dot{V}_s}^2 & 0 & 0 \\ \hline & & & T^2 \cdot \Sigma_{\dot{\theta}_s}^2 & \\ & & & & T^2 \cdot \Sigma_{\dot{F}_0}^2 \end{array} \right] & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \quad (6.7)$$

The first two terms involve state related terms.

The state transition matrix is

$$\underline{\phi} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.8)$$

and the linearized measurement matrix is

$$\tilde{H}(k) = \frac{\partial \underline{C}}{\partial \underline{X}} \bigg|_{\underline{X}'(k)}$$

$$= \begin{bmatrix} \frac{\partial \theta(k)}{\partial R_{cpa}} & \frac{\partial \theta(k)}{\partial X_{cpa}} & \frac{\partial \theta(k)}{\partial V_s} & \frac{\partial \theta(k)}{\partial \theta_s} & \frac{\partial \theta(k)}{\partial F_o} \\ \frac{\partial f(k)}{\partial R_{cpa}} & \frac{\partial f(k)}{\partial X_{cpa}} & \frac{\partial f(k)}{\partial V_s} & \frac{\partial f(k)}{\partial \theta_s} & \frac{\partial f(k)}{\partial F_o} \end{bmatrix} \quad (6.9)$$

$$\tilde{H}(k) = \begin{bmatrix} -\frac{A \cdot X_{cpa}'(k)}{R'(k)^2} & \frac{A \cdot X_{cpa}(k)}{R'(k)^3} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \frac{f'(k)^2 \cdot V_s'(k)}{F_o'(k) \cdot V_p} \cdot \frac{X_{cpa}'(k) \cdot R_{cpa}(k)}{R'(k)^3} & -\frac{f(k)^2 \cdot V_s'(k)}{F_o'(k) \cdot V_p} \cdot \frac{R_{cpa}'(k)}{R'(k)^3} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ -\frac{f'(k)^2 \cdot X_{cpa}'(k)}{F_o'(k) \cdot V_p} \cdot \frac{X_{cpa}'(k)}{R'(k)} & \frac{1}{0} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix} \quad (6.10)$$

where $\underline{X}'(k)$ represents the known track if the analytical equations are used and the predicted states ($\hat{\underline{X}}(k/k-1)$) in the actual extended-Kalman filter. $f'(k)$ is the predicted frequency measurement in the extended-Kalman filter and the

true frequency measurement in the analytical equations.

$R'(k)$ is defined as

$$R'(k) = \sqrt{R'_{cpa}(k)^2 + X'_{cpa}(k)^2} \quad (6.11)$$

and the covariance of measurement noise matrix is

$$R = \begin{bmatrix} \sigma_\theta^2 & 0 \\ 0 & \sigma_f^2 \end{bmatrix} \quad (6.12)$$

$$= \begin{bmatrix} 7.61543 \times 10^{-3} & 0 \\ 0 & 1.6 \times 10^{-3} \end{bmatrix}$$

The filter was simulated using the analytical equations and the Monte-Carlo simulation with the initial conditions given below. The time between measurements was assumed to be 100 seconds.

$$\hat{\underline{X}}(0/-1) = \begin{bmatrix} 3.15 \times 10^3 \text{ yds.} \\ -2.61 \times 10^3 \text{ yds.} \\ 3.38 \text{ yds/sec} \\ 5.30213 \text{ Rad.} \\ 500.432 \text{ Hz.} \end{bmatrix} \quad (6.13)$$

The $\underline{P}(0/-1)$ matrix is

$$\underline{P}(0/-1) = \begin{bmatrix} 1.6245 \times 10^7 & 0 & 0 & 0 & 0 \\ 0 & 4.057 \times 10^6 & 0 & 0 & 0 \\ 0 & 0 & 2.853 & 0 & 0 \\ 0 & 0 & 0 & 0.596 & 0 \\ 0 & 0 & 0 & 0 & 0.684 \end{bmatrix} \quad (6.14)$$

and the true target initial conditions are

$$\underline{X}(0) = \begin{bmatrix} 2992.59 \text{ Yds.} \\ -5295.85 \text{ Yds.} \\ 4.504 \text{ Yds/sec.} \\ 5.06145 \text{ Rad.} \\ 500 \text{ Hz.} \end{bmatrix} \quad (6.15)$$

These initial conditions have been obtained from a method given in [4].)

The analytical equations predict the mean of estimation error to be oscillatory. Monte-Carlo simulation results have indicated that the filter was unstable. The estimation error becomes unbounded as the target passes through CPA. The reason is that the system has two angle measurement equations used depending on the sign of X_{cpa} (see Equation (6.5)). Since the filter has significant estimation error up to CPA, the filter "crosses" the CPA at a different measurement time than the track does. Thus, at a certain instant the filter uses a different angle prediction equation

then the actual measurement equation. This results in a large residual in the estimation equation and the filter begins to diverge. If the filter has estimates with very small error before CPA, the estimator and the target would "cross" the CPA in the same sampling interval. This is a very difficult task because the geometry of the system indicates that any fluctuations in the target heading largely affect the R_{cpa} and X_{cpa} .

In order to force the sign of $\hat{X}_{cpa}(k/k-1)$ to change in the same sampling interval as the track, frequency measurements were tested to determine whether they were "up doppler" or "down doppler", then the sign of the $\hat{X}_{cpa}(k/k-1)$ was reversed accordingly so that X_{cpa} and $\hat{X}_{cpa}(k/k-1)$ have the same sign.

The results of this approach are illustrated in Figures 6.2 through 6.11 for 1000 Monte-Carlo runs shown as the continuous curves. The triangles represent the results obtained from the analytical equations.

Comparing the results of the two methods, one can see that they are completely different (especially the covariances). There are several possible reasons for this, including:

(1) The extended-Kalman filter uses the estimates and predictions for calculation of the $\hat{Q}(k)$, $\hat{H}(k)$ and $\hat{Q}(k)$ matrices, whereas the true target track was used in the analytical equations. Thus, it is reasonable that the gain schedules will differ largely if the estimates are poor.

(2) Since the linearized measurement matrix $\underline{H}(k)$ has been used, the analytical equations do not include the time sharing angle measurement equations; as far as the analytical equations are concerned there is only one angle measurement equation, because the two equations yield the same derivative.

(3) Finally, the analytical equations do not "see" the change that has been made in the filter algorithm to force the filter to cross the CPA in the same measurement interval with the target in the Monte-Carlo simulation.

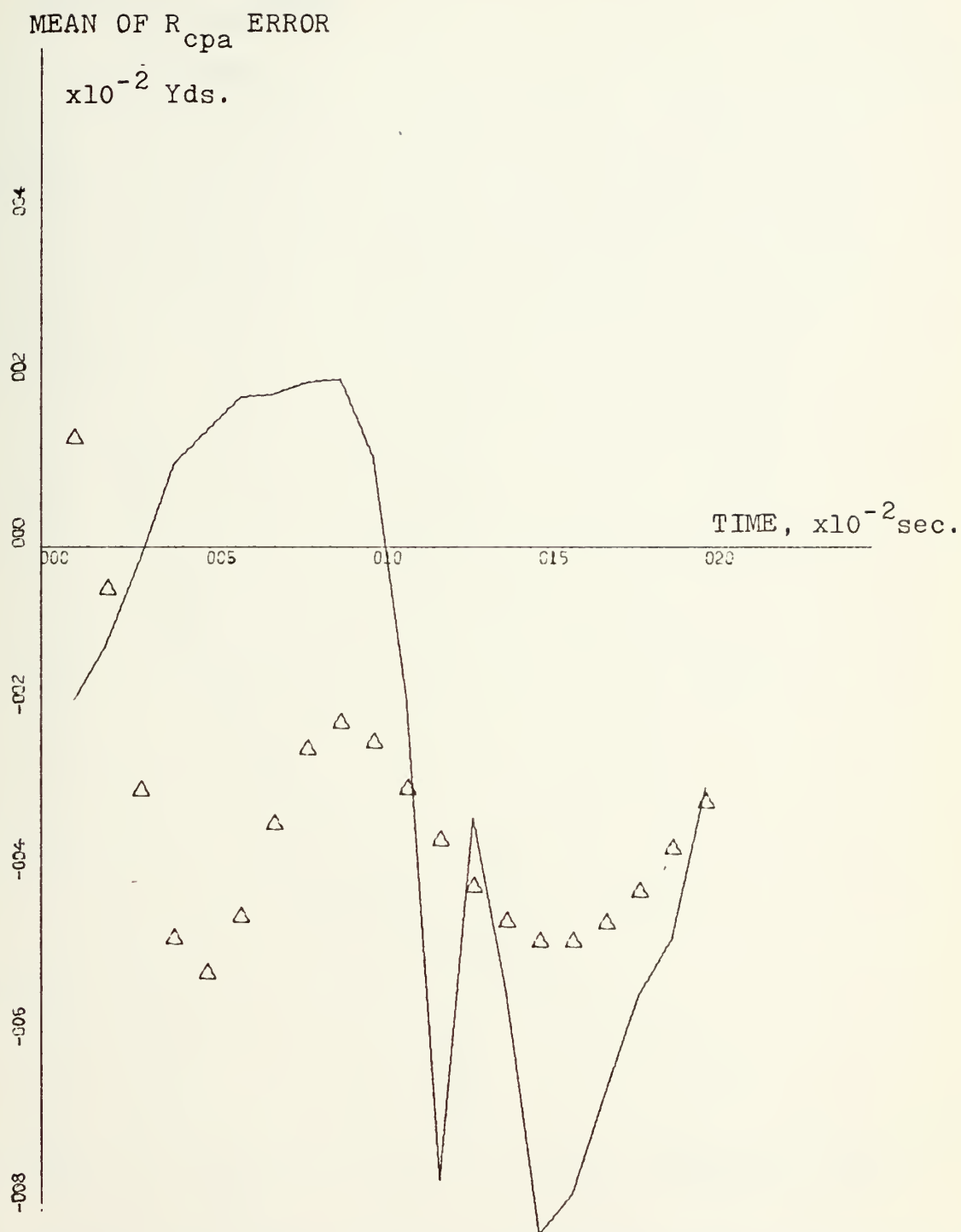


FIG. 6.2. Time history of the mean R_{cpa} estimation error.

X-SCALE:-5.00E+00 UNITS INCH.

Y-SCALE:-2.00E+02 UNITS INCH.



FIG. 6.3. Time history of the mean X_{cpa} estimation error.

X-SCALE=5.00E+00 UNITS INCH.

Y-SCALE=5.00E+02 UNITS INCH.

MEAN V_s ESTIMATION

ERROR

$\times 10^{-1}$ Yds/sec.

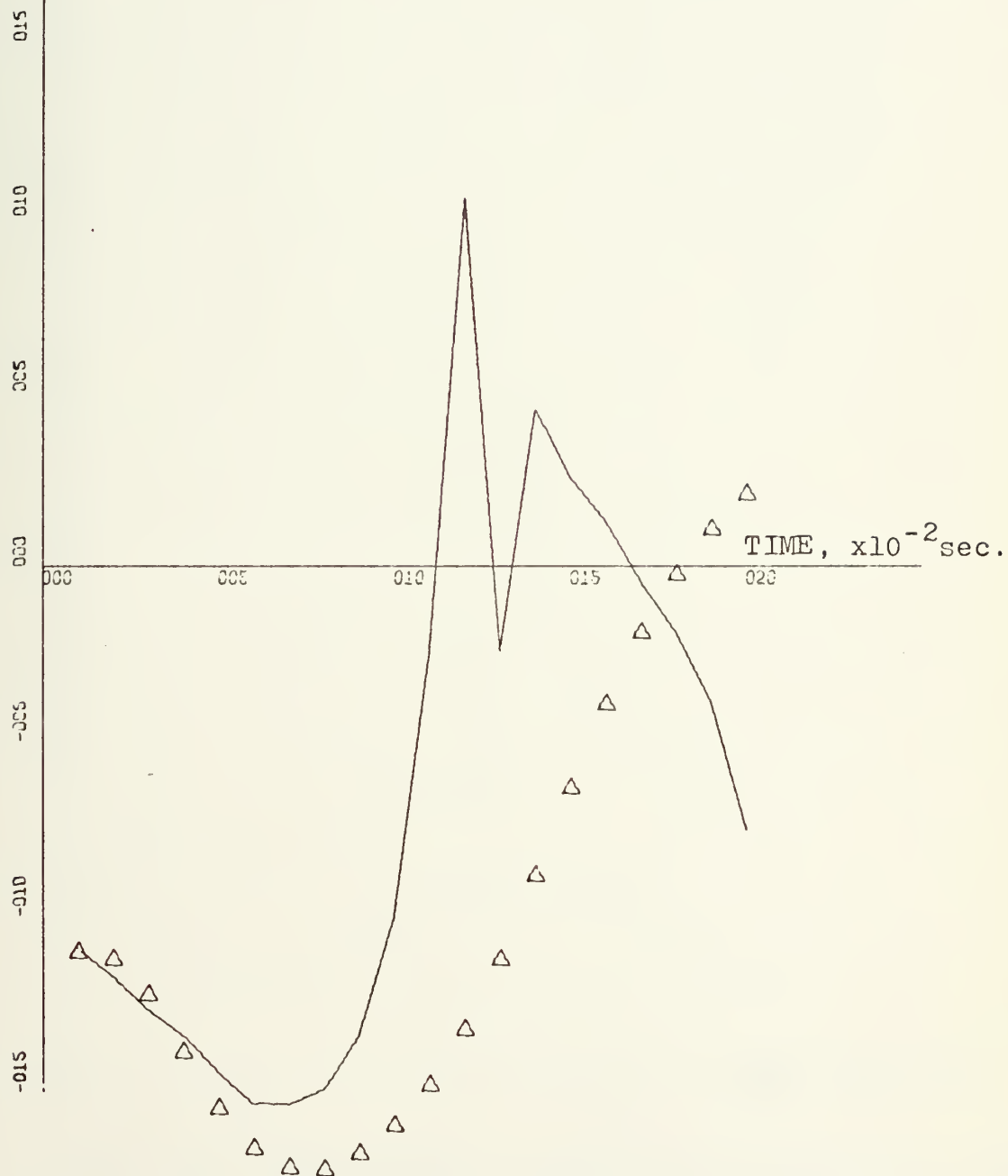


FIG. 6.4. Time history of the mean V_s estimation error.

X-SCALE:-5.00E+00 UNITS INCH.

Y-SCALE:-5.00E-01 UNITS INCH.



FIG. 6.5. Time history of the mean θ_s estimation error.

X-SCALE=5.00E+00 UNITS INCH.

Y-SCALE=2.00E-01 UNITS INCH.

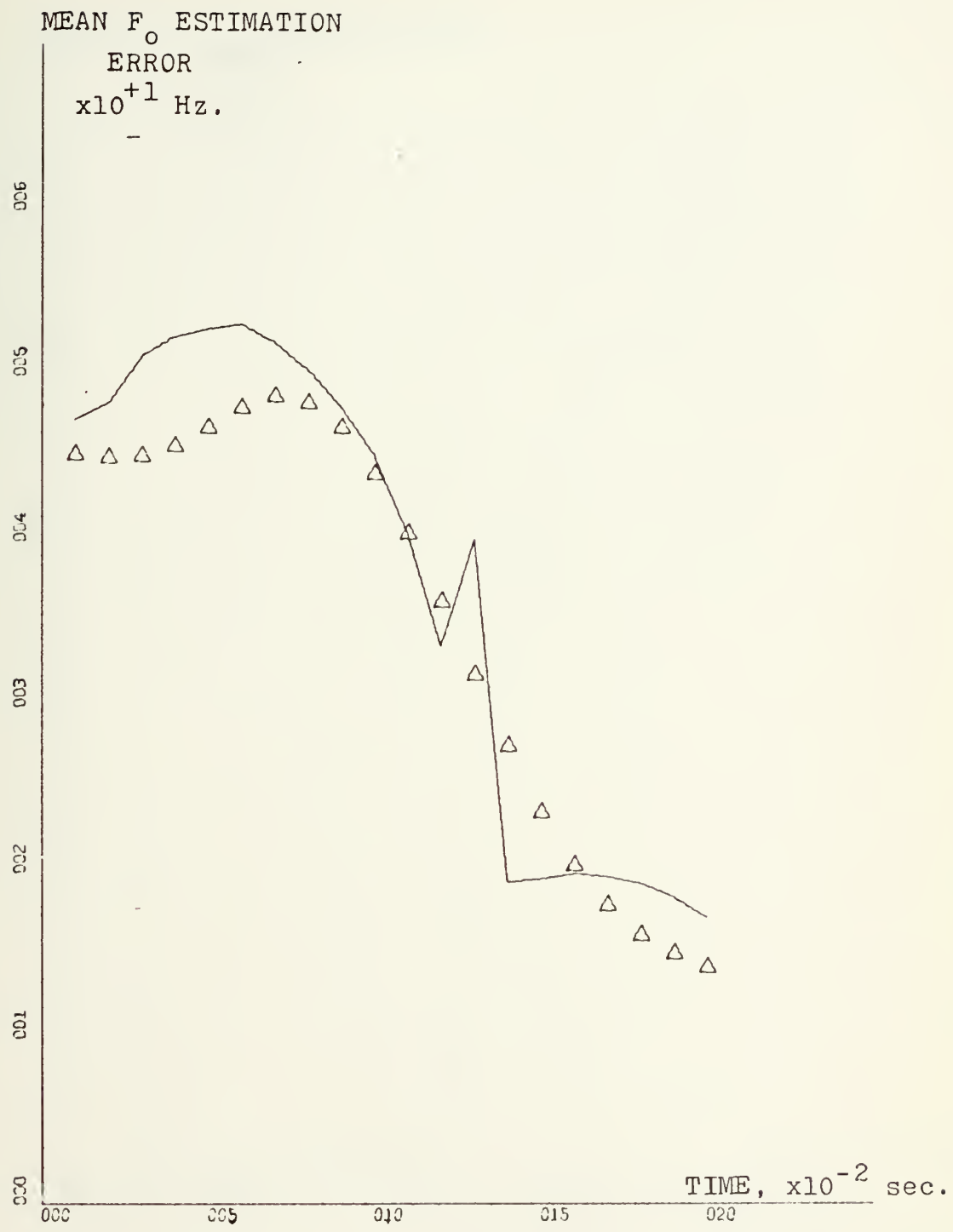


FIG. 6.6. Time history of the mean F_0 estimation error.

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=1.00E-01 UNITS INCH.

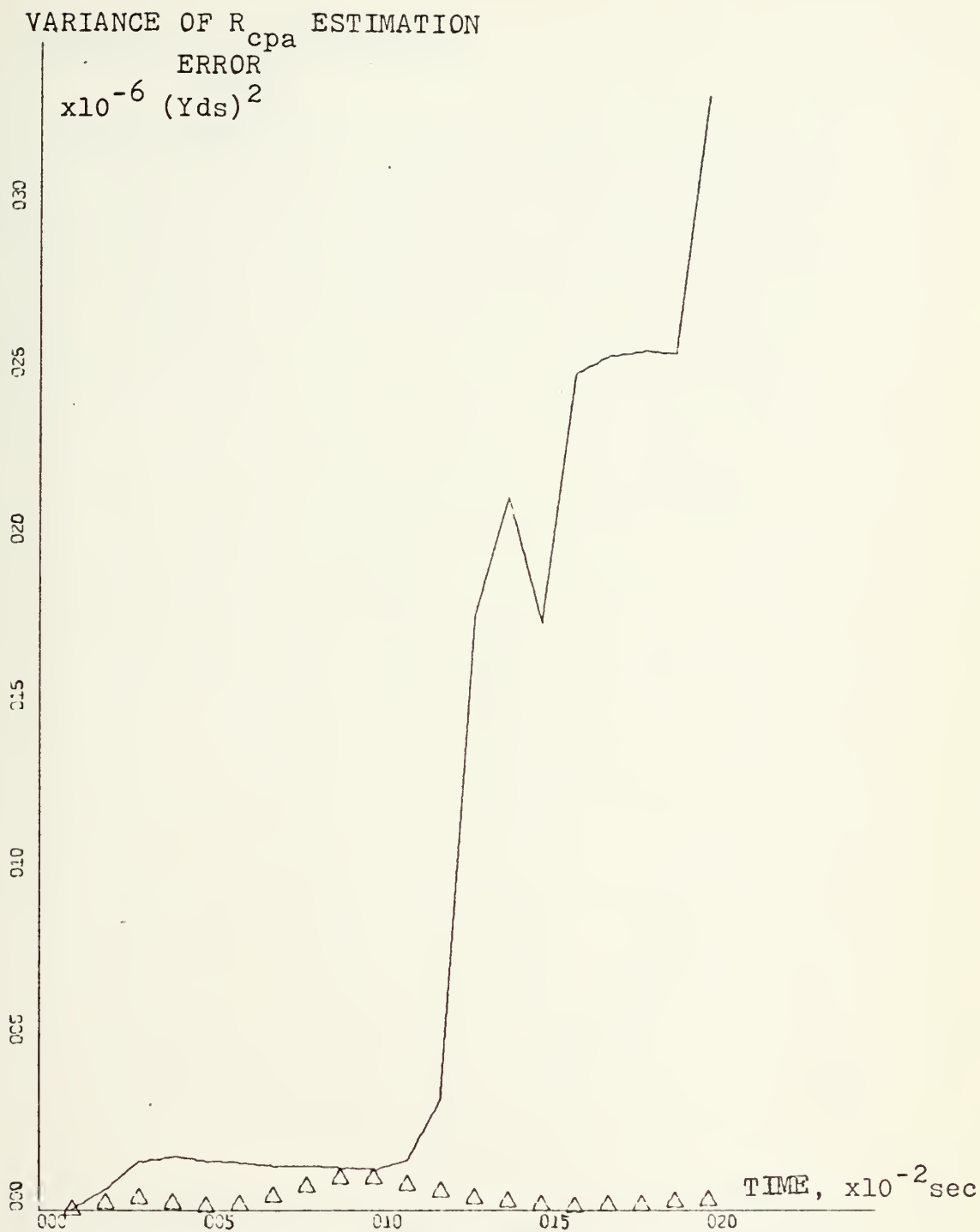


FIG. 6.7. Variance of the mean R_{cpa} estimation error
vs. time.

X-SCALE=5.00E+00 UNITS INCH.

Y-SCALE=5.00E+06 UNITS INCH.

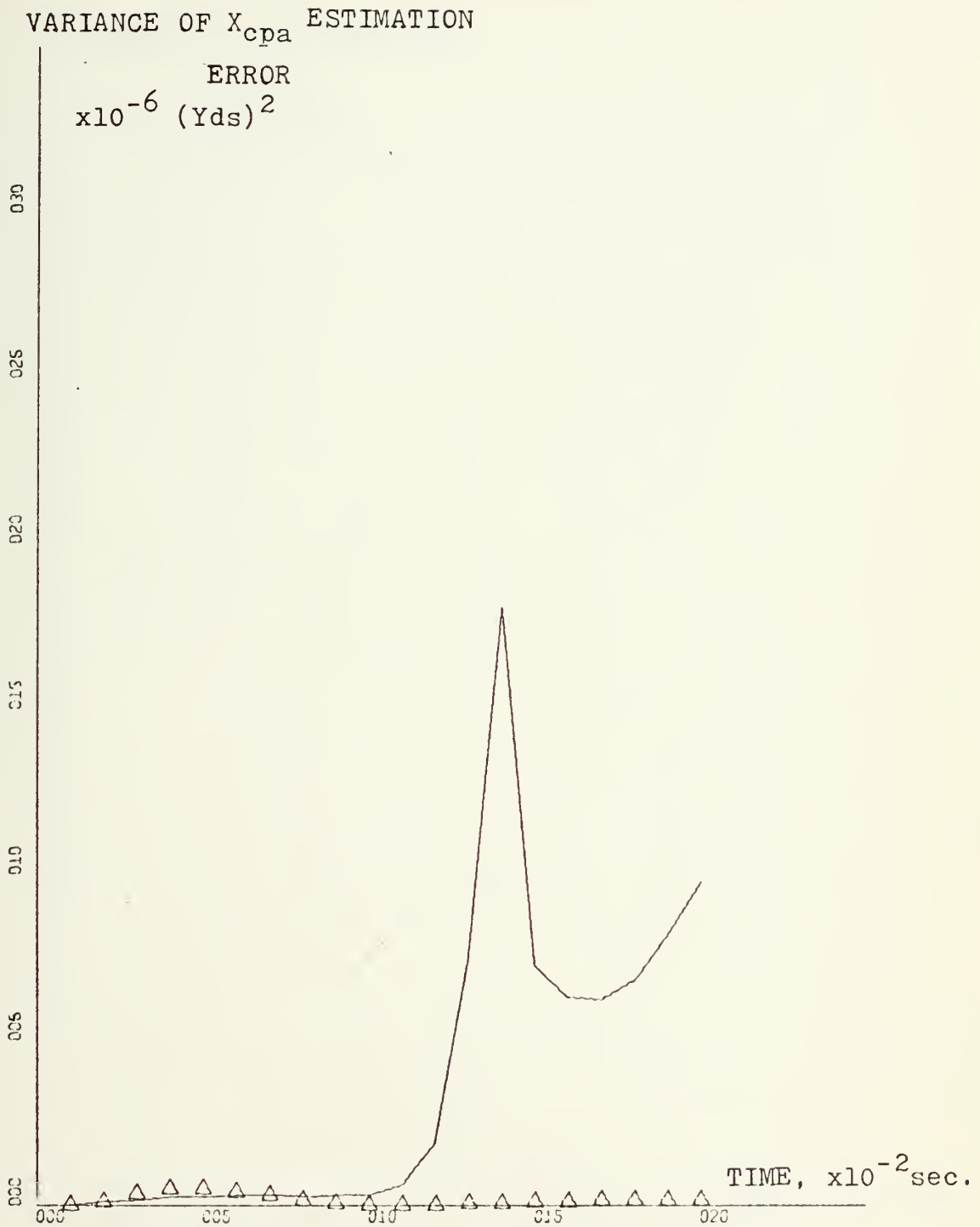


FIG. 6.8. Variance of the mean X_{cpa} estimation error
 vs. time.

X-SCALE:-5.00E+00 UNITS INCH.

Y-SCALE:-5.00E+06 UNITS INCH.



FIG. 6.9. Variance of the mean V_s estimation error vs. time.

X-SCALE= 5.00×10^0 UNITS INCH.

Y-SCALE= 5.00×10^1 UNITS INCH.

VARIANCE OF θ_s ESTIMATION

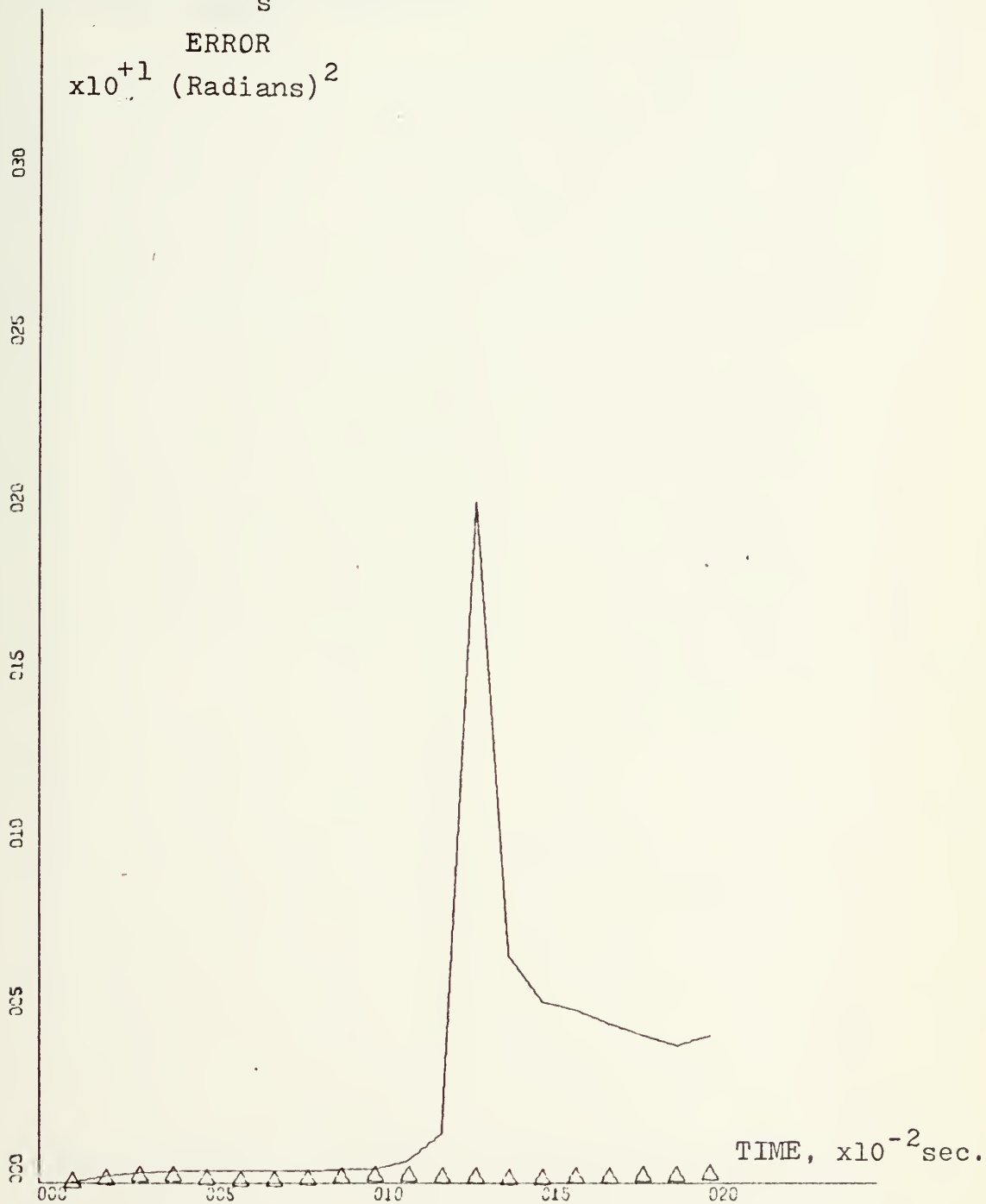


FIG. 6.10. Variance of the mean θ_s estimation error vs. time.

X-SCALE=5.00E+00 UNITS INCH.

Y-SCALE=5.00E-01 UNITS INCH.

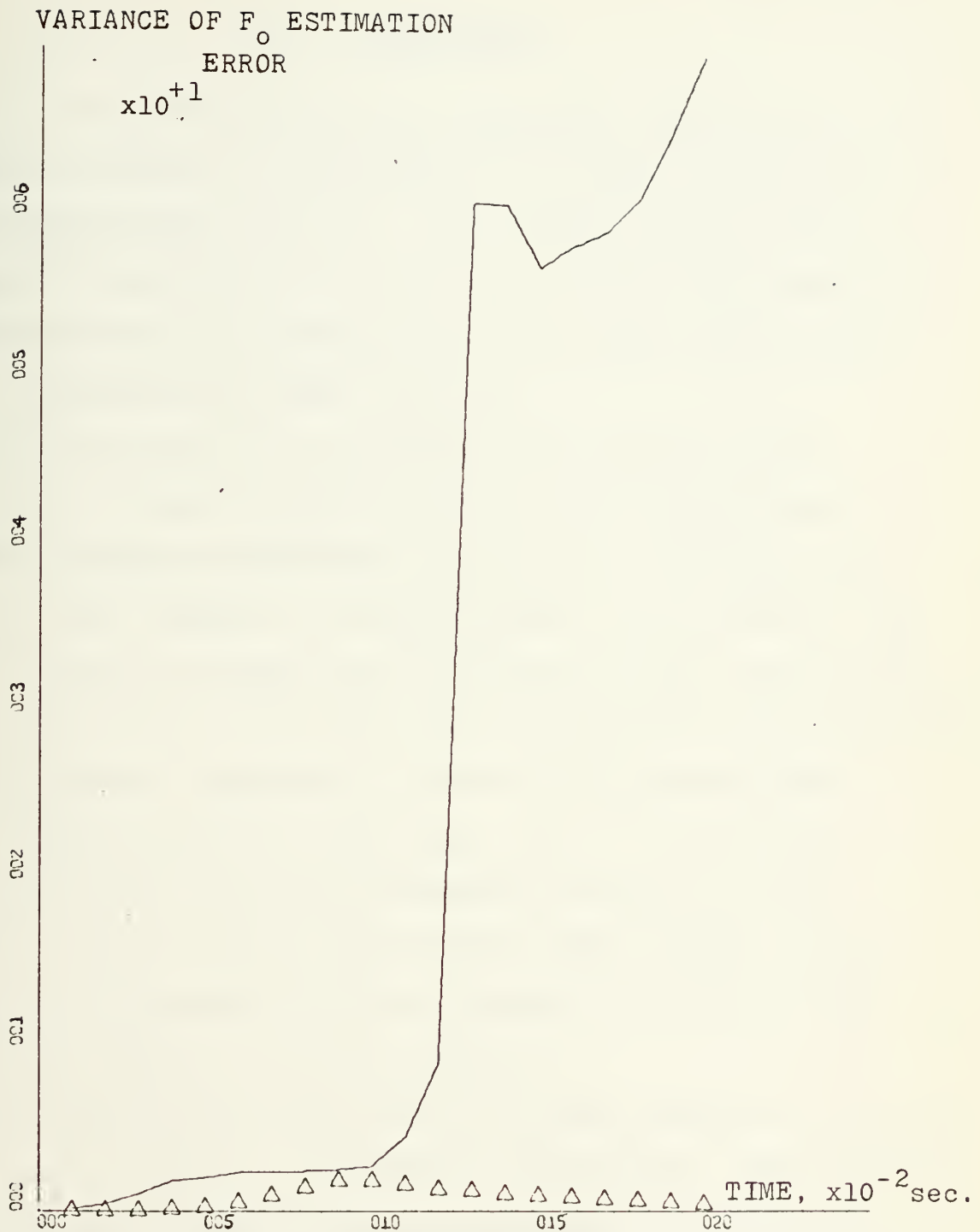


FIG. 6.11. Variance of the mean F_0 estimation error
vs. time.

X-SCALE=5.00E+00 UNITS INCH.

Y-SCALE=1.00E-01 UNITS INCH.

VII. CONCLUSIONS

It has been shown that the analytical equations can be used to evaluate the performance of a linear filter with a pre-computed gain schedule. The results obtained from the analytical equations are exact and are only closely approximated by Monte-Carlo simulations with large ensemble sizes and correspondingly large CPU times.

For the case of the extended-Kalman filter, it is possible to obtain analytical results which approximate those obtained by Monte-Carlo simulations. How close the results are, depends on the performance of the filter. The main difference between the two methods is that one uses the true track whereas the other uses the estimates and predictions as the nominal trajectory to evaluate the matrices $\hat{G}(k)$, $\hat{H}(k)$ (and $\hat{Q}(k)$, $\hat{R}(k)$ in some cases, too). If the filter performs poorly and gives diverging estimates, one can expect that the matrices $\hat{G}(k)$, $\hat{H}(k)$, $\hat{Q}(k)$ and $\hat{R}(k)$ will be completely different in the two methods. This will cause different results in the two methods.

The matrix $\hat{Q}(k)$ effects only the gain schedule. If it is a constant matrix or independent of the states, there will be no difference between results obtained by the analytical equations and Monte-Carlo simulation. The matrix $\hat{R}(k)$ has two kinds of effects. One is similar to that which $\hat{Q}(k)$ has, the other one is that the $\hat{R}(k)$ matrix represents the covariance

of measurement noise, and in Chapter V, it has been shown that the "larger" $R(k)$ is the, larger the difference between results becomes. From the results given in Chapter V, it is observed that the initial filter states effect the results too. Depending on how close the initial filter states are to the true initial states, the time at which the filter converges will change.

In summary, one can use the analytical equations to approximately predict the performance of an extended-Kalman filter. The accuracy of prediction depends on the nature of the problem and the filter.

APPENDIX A

COMPUTER PROGRAMS

A. PROGRAM "EVAL"

The Program "EVAL" is a computer algorithm which solves the analytical equations derived in Chapter II. The algorithm has been written for the special case of the analytical equations which is developed for the MK. 86 Fire Control System as discussed in Reference [1]. The estimation equation has been assumed as

$$\begin{aligned} \hat{\underline{X}}(k/k) = \hat{\underline{X}}(k/k-1) + \underline{G}(k) \left[\underline{M}_0(k) \underline{Z}(k) + \underline{M}_1(k) \underline{Z}(k-1) \right. \\ \left. + \underline{M}_2(k) \underline{Z}(k-2) - \underline{C} \hat{\underline{X}}(k/k-1) \right] \quad (A.1) \end{aligned}$$

and the measurement equation as

$$\underline{Z}(k) = \underline{H} \underline{X}(k) + \underline{V}(k) \quad (A.2)$$

where \underline{H} is the true measurement matrix \underline{C} is the measurement matrix with synthetic measurements. The matrices $\underline{M}_0(k)$, $\underline{M}_1(k)$, $\underline{M}_2(k)$ are measurement weighting matrices which have been used in the MK. 86 Fire Control System.

In the development of this research these matrices are defined as

$$\underline{M}_0(k) = \underline{I} \quad (\text{qxq Identity matrix}) \quad (A.3)$$

$$\underline{M}_1(k) = \underline{M}_2(k) = \underline{0} \quad (A.4)$$

$$\underline{C} = \underline{H} \quad (A.5)$$

Then Equation (A.1) reduces to the simpler form

$$\hat{\underline{X}}(k/k) = \hat{\underline{X}}(k/k-1) + \underline{G}(k) \left[\underline{Z}(k) - \underline{C} \hat{\underline{X}}(k/k-1) \right] \quad (\text{A.6})$$

With the assumptions given above and starting with the definition of the estimation error, i.e.

$$\tilde{\underline{X}}(k) = \hat{\underline{X}}(k/k) - \underline{X}(k) \quad (\text{A.7})$$

(where $\tilde{\underline{X}}(k)$ is the true estimation error vector)

the following analytical results have been derived in [1]:

1. Covariance of estimation error

$$\begin{aligned} \tilde{\underline{P}}(k/k) = & \underline{S}_3(k) \tilde{\underline{P}}(k-1/k-1) \underline{S}_3(k)^T + \underline{S}_0(k) \underline{R}(k) \underline{S}_0(k)^T \\ & + \underline{S}_1(k) \underline{R}(k-1) \underline{S}_1(k)^T + \underline{S}_2(k) \underline{R}(k-2) \underline{S}_2(k)^T \\ & + \underline{A}_1(k) + \underline{A}_1(k)^T + \underline{A}_2(k) + \underline{A}_2(k)^T \end{aligned} \quad (\text{A.8})$$

2. Mean of estimation error

$$\tilde{\underline{\mu}}(k) = \underline{S}_3(k) \tilde{\underline{\mu}}(k-1) + \underline{D}(k) + \underline{S}_3(k) \underline{X}(k-1) \quad (\text{A.9})$$

3. Initializing values

$$\tilde{\underline{P}}(0/0) = \underline{S}_0(0) \underline{R}(0) \underline{S}_0(0)^T \quad (\text{A.10})$$

$$\tilde{\underline{\mu}}(0) = \left[\underline{I} - \underline{G}(0) \underline{C} \right] \hat{\underline{X}}(0/-1) - \left[\underline{I} - \underline{G}(0) \underline{M}_0(0) \underline{H} \right] \underline{X}(0) \quad (\text{A.11})$$

4. Mean of N-step prediction error

$$\tilde{\underline{\mu}}_p(N+k/k) = \phi^N \tilde{\underline{\mu}}(k) + \phi^N \underline{X}(k) - \underline{X}(N+k) \quad (\text{A.12})$$

5. Covariance of N-step prediction error

$$\tilde{\mathbf{P}}(N+k/k) = \phi^N \tilde{\mathbf{P}}(k/k) [\phi^N]^T \quad (\text{A.13})$$

where

$$\mathbf{S}_0(k) = \mathbf{G}(k) \mathbf{M}_0(k)$$

$$\mathbf{S}_1(k) = \mathbf{G}(k) \mathbf{M}_1(k)$$

$$\mathbf{S}_2(k) = \mathbf{G}(k) \mathbf{M}_2(k)$$

$$\mathbf{S}_3(k) = [\mathbf{I} - \mathbf{G}(k) \mathbf{C}] \phi$$

$$\mathbf{A}_1(k) = \mathbf{S}_3(k) \mathbf{S}_0(k-1) \mathbf{R}(k-1) \mathbf{S}_1(k)^T$$

$$\mathbf{A}_2(k) = \mathbf{S}_3(k) [\mathbf{S}_3(k-1) \mathbf{S}_0(k-2) + \mathbf{S}_1(k-1)] \mathbf{R}(k-1) \mathbf{S}_2(k)^T$$

$$\mathbf{D}(k) = [\mathbf{S}_0(k) \mathbf{H} - \mathbf{I}] \mathbf{X}(k) + \mathbf{S}_1(k) \mathbf{H} \mathbf{X}(k-1) + \mathbf{S}_2(k) \mathbf{H} \mathbf{X}(k-1)$$

With the assumptions given by Equations (A.3) - (A.5),

Equations (A.8) through (A.13) reduce to the form of

equations derived in Chapter II.

The computer program given in [1] has been used after making some small changes and adding three subroutines. The program listing is given at the end of this Appendix. In order to use the program, one must provide subroutine AK which computes the linearized state transition matrix $\phi(k)$, subroutine H(k) which calculates the matrix HKK and subroutine UK which calculates the vector $\mathbf{U}'(k-1)$. If the matrices $\mathbf{Q}(k)$ and $\mathbf{R}(k)$ are required to be calculated on-line,

then subroutines QON and RON must also be provided. It is also necessary to give the proper values to certain flags for required operation.

B. MONTE-CARLO SIMULATION

A listing of the program used for Monte-Carlo simulation is also given at the end of this Appendix. The program has been set up for use with both linear and extended-Kalman filters. One can simulate a linear or extended-Kalman filter by selecting the proper flags and supplying the proper subroutines. For extended-Kalman filter simulation, one must provide subroutine MEAS for simulation of measurements, subroutine TRACK for generating the target track (if not read in), subroutine HKK for calculation of $H(k)$, subroutine AK for calculation of $\phi(k)$, subroutine CK for calculation of $C(\underline{X}(k/k-1))$, subroutine XPRED for calculation of $\underline{X}(k+1/k) = \underline{f}(\underline{X}(k/k))$, and subroutines QON and RON for $Q(k)$ and $R(k)$ if they are not read in.


```

C      IH=0  H MATRIX WILL BE COMPUTED ONLINE
C      IQ=0  Q MATRIX WILL BE COMPUTED ONLINE
C      IPI=0  TWO TRACKS ARE GOING TO BE USED  WITH GIVEN PROB
C
C      READ (5,55) N,L,NA,INOISE,IG,ITRK,NSAM,ND
C      READ(5,55) IP,IPI,IH,IQ
C      IF(IPI.EQ.0) READ (5,56) P
C      IPI=1
C      FORM NXN IDENTITY MATRIX EID
C
C      DO 2 I=1,N
C
C      DO 2 J=1,N
C      IF (I.EQ.J) GO TO 1
C      EID(I,J) = 0.00
C      GO TO 2
C      1 EID(I,J) = 1.00
C      2 CONTINUE
C
C      READ (5,56) (XHIT(I),I=1,N)
C
C      DO 3 I=1,L
C      3 READ (5,56) (XOK(I,J),J=1,NA)
C
C      DO 4 I=1,L
C      4 READ (5,56) (XIK(I,J),J=1,NA)
C
C      DO 5 I=1,L
C      5 READ (5,56) (X2K(I,J),J=1,NA)
C      IF (IP.EQ.0) GO TO 6
C
C      DO 6 I=1,N
C      6 READ (5,56) (P-I(I,J),J=1,N)
C      CONTINUE
C
C      IF (IH.EQ.0) GO TO 8
C
C      DO 7 I=1,L
C      7 READ (5,56) (C(I,J),J=1,N)
C
C      DO 8 I=1,NA
C      8 READ (5,56) (H(I,J),J=1,N)
C      CONTINUE

```

EVAL00043
EVAL00044

EVAL00045
EVAL00046
EVAL00047
EVAL00048
EVAL00049
EVAL00050
EVAL00051
EVAL00052
EVAL00053
EVAL00054
EVAL00055
EVAL00056
EVAL00057
EVAL00058
EVAL00059
EVAL00060
EVAL00061
EVAL00062
EVAL00063
EVAL00064
EVAL00065
EVAL00066
EVAL00067
EVAL00068

EVAL00069
EVAL00070

EVAL00072

EVAL00073
EVAL00074
EVAL00075
EVAL00076
EVAL00077
EVAL00078

EVAL0080
EVAL0082
EVAL0083
EVAL0084
EVAL0085
EVAL0086
EVAL0087
EVAL0088
EVAL0089
EVAL0090
EVAL0091
EVAL0092
EVAL0093
EVAL0094
EVAL0095

EVAL0096
EVAL0097
EVAL0098
EVAL0099
EVAL0100
EVAL0101
EVAL0102

EVAL0104
EVAL0105
EVAL0106
EVAL0107
EVAL0108
EVAL0109
EVAL0110
EVAL0111
EVAL0112
EVAL0113
EVAL0114
EVAL0115
EVAL0116
EVAL0118
EVAL0119
EVAL0120
EVAL0121

```

C      85 WRITE (6,57) N,L,NA,INOISE,NSAM
      WRITE (6,58)
      IF (IG.EQ.0) GO TO 10
C
C      DO 9 K=1,NSAM
C
C      DO 9 I=1,N
C      9 READ (5,59) (GKS(I,J,K),J=1,L)
C      GO TO 16
C
C      DO 11 I=1,N
C      11 READ (5,56) (PK(I,J),J=1,N)
C      P(0/-1) READ IN TO ARRAY PK FOR GAIN CALCULATION.
C      WRITE (6,83)
C
C      DO 84 I=1,N
C      84 WRITE (6,63) (PK(I,J),J=1,N)
C
C      IF(IQ.EQ.0) GO TO 13
C
C      DO 12 I=1,N
C      12 READ (5,56) (RKM2(I,J),J=1,N)
C      WRITE (6,60)
C
C      DO 13 I=1,N
C      WRITE (6,63) (RKM2(I,J),J=1,N)
C      13 CONTINUE
C
C      Q READ IN TO ARRAY RKM2 FOR GAIN CALCULATION
C
C      DO 14 I=1,NA
C      14 READ (5,56) (RKM1(I,J),J=1,NA)
C      WRITE (6,61)
C
C      DO 15 I=1,NA
C      15 WRITE (6,63) (RKM1(I,J),J=1,NA)
C
C      R USED FOR GAIN CALCULATION READ IN AND STORED IN RKM1. NOTE THAT
C      R IS ASSUMED TO BE NA X NA.
C      16 IF (ITRK.EQ.1) GO TO 18
C
C      DO 17 K=1,NSAM
C      17 READ (5,59) (XKS(I,K),I=1,N)

```


EVAL01122
EVAL01123
EVAL01124
EVAL01125
EVAL01126
EVAL01127

EVAL01128
EVAL01129
EVAL01130
EVAL01131
EVAL01132
EVAL01133
EVAL01134
EVAL01135
EVAL01136
EVAL01137
EVAL01138
EVAL01139
EVAL01140
EVAL01141
EVAL01142
EVAL01143
EVAL01144
EVAL01145
EVAL01146
EVAL01147
EVAL01148
EVAL01149
EVAL01150
EVAL01151
EVAL01153
EVAL01154
EVAL01155
EVAL01156
EVAL01157
EVAL01158
EVAL01159
EVAL01160
EVAL01161
EVAL01162
EVAL01163
EVAL01164
EVAL01165

```

C      GO TO 19
C      CALL TRACK
C      THE ARRAY CON IS FOR STORAGE OF CONSTANTS WHICH MAY BE NEEDED
C      IN SUBROUTINES.
18  READ (5,56) (CON(I),I=1,8)
19  IF(IP.EQ.0) CALL AK
    IF(IH.EQ.0) CALL HKK
    IF(IQ.EQ.0) CALL QON
    IF(IC.EQ.0) CALL GAIN
    IF (INOISE.NE.0) GO TO 22

C      DO 20 I=1,NA
C      20 READ (5,56) (RK(I,J),J=1,NA)
C
C      DO 21 I=1,NA
C
C      DO 21 J=1,NA
C      RKM1(I,J) = RK(I,J)
C      RKM2(I,J) = RK(I,J)
C
C      21 RKM2(I,J) = RK(I,J)
C      22 CCNINUE
C      WRITE OUT PARTS OF INPUT DATA
C      WRITE (6,62)
C
C      DO 23 I=1,N
C      23 WRITE (6,63) (GKS(I,J,1),J=1,L)
C
C      DO 24 I=1,N
C      24 WRITE (6,63) (GKS(I,J,NSAM),J=1,L)
C
C      IF (IG.NE.0) GO TO 27
C      WRITE (6,64)
C
C      DO 25 I=1,N
C      25 WRITE (6,63) (PKS(I,J,1),J=1,N)
C
C      DO 26 I=1,N
C      26 WRITE (6,63) (PKS(I,J,NSAM),J=1,N)
C
C      27 WRITE (6,65)
C      WRITE (6,63) (XKS(I,1),I=1,N)
C      WRITE (6,63) (XKS(I,NSAM),I=1,N)
C      WRITE (6,66)
C      WRITE (6,63) (XHIT(I),I=1,N)

```


C	IF(IP.EQ.0) GO TO 28	EVAL0166
	WRITE (6,67)	EVAL0167
	DO 28 I=1,N	EVAL0168
	WRITE (6,63) (PHI(I,J),J=1,N)	
	CONTINUE	
28	IF(IH.EQ.0) GO TO 86	EVAL0170
C	WRITE (6,68)	EVAL0171
C		EVAL0172
	DO 29 I=1,NA	EVAL0173
	WRITE (6,63) (H(I,J),J=1,N)	EVAL0174
C	WRITE (6,69)	EVAL0175
C		EVAL0176
	DO 30 I=1,L	EVAL0177
	WRITE (6,63) (C(I,J),J=1,N)	EVAL0178
30		EVAL0179
C	WRITE (6,70)	EVAL0180
C		EVAL0182
	DO 31 I=1,L	EVAL0183
	WRITE (6,63) (MOK(I,J),J=1,NA)	EVAL0184
C	WRITE (6,71)	EVAL0185
C		EVAL0186
	DO 32 I=1,L	EVAL0187
	WRITE (6,63) (M1K(I,J),J=1,NA)	EVAL0188
32		EVAL0189
C	WRITE (6,72)	EVAL0190
C		EVAL0191
	DO 33 I=1,L	EVAL0192
	WRITE (6,63) (M2K(I,J),J=1,NA)	EVAL0193
33		EVAL0194
C	IF (INOISE.NE.0) GO TO 35	EVAL0195
	WRITE (6,73)	EVAL0196
C		EVAL0197
	DO 34 I=1,NA	EVAL0198
	WRITE (6,63) (RK(I,J),J=1,NA)	EVAL0199
34		EVAL0200
C	CONTINUE	EVAL0201
	WRITE (6,74)	EVAL0202
C	COMPUTATION	EVAL0203
C	BEGINS HERE.	EVAL0204
C	SETTINGS FOR SOME OF THE MATRICES INVOLVED AND SO ARE	EVAL0205
C	DONE OUTSIDE OF THE MAIN LOOP.	EVAL0206
C		EVAL0207
C		EVAL0208
C	COMPUTE SO(0)=SOK	EVAL0209
C		EVAL0210

IF MEANS AND COVARIANCES OF ESTIMATION ERROR
THE FIRST TWO SAMPLES REQUIRE SOME ATYPICAL
BEGINNING HERE.
SETTINGS FOR SOME OF THE MATRICES INVOLVED AND SO ARE
DONE OUTSIDE OF THE MAIN LOOP.


```

K = 1 LOADG
CALL PROD (GK,MOK,N,L,NA,SOK)
SET S1(O),S2(O),A1(O),A2(O),S3(O) TO ZERO SO THAT SUBROUTINE
COVE CAN BE USED TO COMPUTE P(O) AS GIVEN BY EQ. (4.38)
IF (INDISE.NE.O) CALL RNOISE
DO 36 I=1,N
DO 36 J=1,NA
S1K(I,J) = 0.D0
36 S2K(I,J) = 0.D0
DO 37 I=1,N
DO 37 J=1,N
A1K(I,J) = 0.D0
A2K(I,J) = 0.D0
37 S3K(I,J) = 0.D0
CALL COVE
P(O) HAS BEEN COMPUTED
COMPUTE MU(O) -- SUBROUTINE MEANS NOT USED THIS TIME ONLY
DO 38 I=1,N
XK(I) = XKS(I,K)
38 VTMP1(I) = XHIT(I)-XK(I)
IF(IH.NE.O) GO TO 88
DO 88 I=1,NA
DO 88 J=1,N
H(I,J)=HK(I,J,K)
C(I,J)=HK(I,J,K)
88 CONTINUE
CALL VPROD (C,XHIT,L,N,VTMP2)
CALL VPROD (H,XK,NA,N,VTMP3)
CALL VPROD (MOK,VTMP3,L,NA,VTMP3)
CALL VSUB (VTMP3,VTMP2,L,VTMP2)
CALL VPROD (GK,VTMP2,N,L,MUK)
CALL VADD (MUK,VTMP1,N,MUK)
COMPUTATION OF MU(O) COMPLETE
STORE P(O), MU(O)
CALL STORE
CALL UP FOR NEXT CYCLE
K = 2

```

EVAL0211
 EVAL0212
 EVAL0213
 EVAL0214
 EVAL0215
 EVAL0216
 EVAL0217
 EVAL0218
 EVAL0219
 EVAL0220
 EVAL0221
 EVAL0222
 EVAL0223
 EVAL0224
 EVAL0225
 EVAL0226
 EVAL0227
 EVAL0228
 EVAL0229
 EVAL0230
 EVAL0231
 EVAL0232
 EVAL0233
 EVAL0234
 EVAL0235
 EVAL0236
 EVAL0237
 EVAL0238

EVAL0239
 EVAL0240
 EVAL0241
 EVAL0242
 EVAL0243
 EVAL0244
 EVAL0245
 EVAL0246
 EVAL0247
 EVAL0248
 EVAL0249
 EVAL0250


```

REPLACE XKMI BY XK, REPLACE SOKMI BY SOK, RESET XK
DO 39 I=1,N
XKMI(I) = XK(I)
XK(I) = XKS(I,K)
DO 39 J=1,NA
39 SOKMI(I,J) = SOK(I,J)
IF (INOISE.EQ.0) GO TO 41
IF INOISE.NE.0 COMPUTE NEW VALUE OF R(K)
DO 40 I=1,NA
DO 40 J=1,NA
40 RKMI(I,J) = RK(I,J)
CALL RNOISE
41 CONTINUE
CALCULATE S0(I)=SOK
CALL LOADG
CALL PROD (GK,MOK,N,L,NA,SOK)
CALCULATE S1(I)=S1K
CALL PROD (GK,M1K,N,L,NA,S1K)
SET S2(I)=0
DO 42 I=1,N
DO 42 J=1,NA
42 S2K(I,J) = 0.D0
CALCULATE S3(I)=S3K
CALL S3
CALCULATE A1(I)=A1K
CALL A1
SET A2(I)=0 BECAUSE S2K=0
DO 43 I=1,N
DO 43 J=1,N
43 A2K(I,J) = 0.D0
NOW READY TO COMPUTE P(I)
WRITE(6,73) (SOK(I,I),I=1,N)
CALL COVE
SET UP TO COMPUTE MU(1)=MUK FROM EQ. (4.34)
SET XKM2 = 0

```



```

DO 44 I=1,N
44 XKM2(I) = 0.00
      CALL MEANE
      CALL MEANE
      WRITE(6,73) (MUK(I),I=1,N)
      STORE P(1) AND MU(1)
      CALL STORE
      BEGIN GENERAL DO LOOP (K=3,NSAMPLES) HERE
      DO 49 K=3,NSAM
      SHIFT LOCATIONS OF MATRICES TO PREPARE FOR NEXT CYCLE
      DO 45 I=1,N
      XKM2(I) = XK(S(I,K-2))
      XKMI(I) = XK(S(I,K-1))
      XK(I) = XK(S(I,K))
      DO 45 J=1,NA
      SOKM2(I,J) = SOKMI(I,J)
      SOKMI(I,J) = SOK(I,J)
      SIKMI(I,J) = SIK(I,J)
45
      DO 46 I=1,N
      DO 46 J=1,N
      S3KMI(I,J) = S3K(I,J)
46
      IF (INOISE.EQ.0) GO TO 48
      CCMPUTE R(K) IF INOISE.NE.0
      FIRST SHIFT THE MATRICES RKM1,RK TO RKM2,RKM1,RESP
      DO 47 I=1,NA
      DO 47 J=1,NA
      RKM2(I,J) = RKM1(I,J)
      RKM1(I,J) = RK(I,J)
47
      CALL RNOISE
      CONTINUE
      CALL LOADG
      CCMPUTE SO(K)=SOK FROM EQ. (4.33)
      CALL PROD (GK,MOK,N,L,NA,SOK)
      COMPUTE S1(K)=SIK FROM EQ. (4.33)
      CALL PROD (GK,M1K,N,L,NA,S1K)

```


EVAL0347
EVAL0348
EVAL0349
EVAL0350
EVAL0351
EVAL0352
EVAL0353
EVAL0354
EVAL0355
EVAL0356
EVAL0357
EVAL0358
EVAL0359
EVAL0360
EVAL0361
EVAL0362
EVAL0363

EVAL0364
EVAL0365
EVAL0366
EVAL0367

```

C      COMPUTE S2(K)=S2K FROM EQ. (4.33)
C      CALL PROD (GK,M2K,N,L,NA,S2K)
C      COMPUTE S3(K)=S3K FROM EQ. (4.33)
C      CALL S3
C      COMPUTE A1(K)=A1K FROM EQ. (4.33)
C      CALL A1
C      COMPUTE A2(K)=A2K FROM EQ. (4.33)
C      CALL A2
C      COMPUTE COVARIANCE OF ESTIMATION ERROR
C      AT TIME K, P(K)=PK
C      CALL COVE
C      COMPUTE D(K)=DK AND MU(K)=MUK
C      CALL D
C      CALL MEANE
C      STORE P(K) AND MU(K)
C      CALL STORE
C      CONTINUE
49      IF(IP1.NE.0) GO TO 93
C      IF(IP2.EQ.2) GO TO 91
C      IP2=2
C      DO 90 K=1,NSAM
C
C      DO 90 I=1,N
C      PMUKS1(I,K)=MUKS(I,K)
C
C      DO 90 J=1,N
C      PKS1(I,J,K)=PKS(I,J,K)
C      CONTINUE
C
C      GO TO 2
C
C      DO 92 K=1,NSAM
C
C      DO 92 I=1,N
C      PMUKS2(I,K)=MUKS(I,K)
C
C      DO 92 J=1,N
C      PKS2(I,J,K)=PKS(I,J,K)
C      CONTINUE
C
C      CALL PROB
C      CONTINUE
C      PRINT OUT RESULTS
C      WRITE (6,75)
C

```



```

DO 51 K=1, NSAM
WRITE (6,79) K
WRITE (6,80) (MUKS(I,K), I=1,N)
WRITE (6,76)
C
DO 50 I=1,N
50 WRITE (6,81) (PKS(I,J,K), J=1,N)
C
51 WRITE (6,82)
C
*****
IF (IPCH.NE.0) MEANS AND COVARIANCES ARE PUNCHED
READ (5,77) IPCH
IF (IPCH.EQ.0) GO TO 54
C
DO 52 K=1, NSAM
52 WRITE (7,78) (MUKS(I,K), I=1,N)
C
DO 53 K=1, NSAM
C
DO 53 I=1,N
53 WRITE (7,78) (PKS(I,J,K), J=1,N)
C
54 CONTINUE
C
PLOT OUTPUT
CALL OPLOT (MUKS,PKS,GKS,NSAM,N,NA)
STOP
C
55 FORMAT (8I10)
56 FORMAT (30X,'A PARTIAL LISTING OF INPUT DATA FOLLOWS',/)
57 FORMAT (2X,N = ,I1,2X,L = ,I1,2X,NA = ,I1,2X,INOISE = ,I1,
58 1 2X,NSAM = ,I3,/)
59 FORMAT (4F20.0)
60 FORMAT (//,2X,THE Q MATRIX USED IN THE GAIN CALCULATION IS,/)
61 FORMAT (//,2X,THE R MATRIX USED IN THE GAIN CALCULATION IS,/)
62 FORMAT (2X,GAIN MATRICES FOR FIRST AND LAST SAMPLES ARE,/)
63 FORMAT (9(2X,1PD12.5),/)
64 FORMAT (//,1X,THE THEORETICAL COV OF EST ERROR MATRIX CALC BY
1 "GAIN" FOR FIRST AND LAST SAMPLE ARE,/)
65 FORMAT (//,1X,STATISTICAL TRAJ. FOR FIRST AND LAST SAMPLE ARE,/)
66 FORMAT (//,1X,INITIAL STATE ESTIMATE IS,/)
67 FORMAT (//,1X,THE PHI MATRIX IS,/)
68 FORMAT (//,1X,THE H MATRIX IS,/)
69 FORMAT (//,1X,THE C MATRIX IS,/)

```

EVAL0368
 EVAL0369
 EVAL0370
 EVAL0371
 EVAL0372
 EVAL0373
 EVAL0374
 EVAL0375
 EVAL0376
 EVAL0377
 EVAL0378
 EVAL0379
 EVAL0380
 EVAL0381
 EVAL0382
 EVAL0383
 EVAL0384
 EVAL0385
 EVAL0386
 EVAL0387
 EVAL0388
 EVAL0389
 EVAL0390
 EVAL0391
 EVAL0392
 EVAL0393
 EVAL0394

EVAL0396
 EVAL0397
 EVAL0398
 EVAL0399
 EVAL0400
 EVAL0401
 EVAL0402
 EVAL0403
 EVAL0404
 EVAL0405
 EVAL0406
 EVAL0407
 EVAL0408
 EVAL0409
 EVAL0410
 EVAL0411
 EVAL0412
 EVAL0413
 EVAL0414


```

70 FORMAT (///,1X,'THE MATRIX MO(K) IS',/)
71 FORMAT (///,1X,'THE MATRIX M1(K) IS',/)
72 FORMAT (///,1X,'THE MATRIX M2(K) IS',/)
73 FORMAT (///,1X,'THE MATRIX R IS',/)
74 FORMAT (11)
75 FORMAT (2X,'MEANS AND COVARIANCES OF EST. ERROR',/)
76 FORMAT (///,1X,'THE COVARIANCE OF EST. ERROR MATRIX IS',/)
77 FORMAT (11)
78 FORMAT (41PD20.12))
79 FCFORMAT (5X,'K =',I3,/)
80 FORMAT (1X,'THE ESTIMATION ERROR MEAN IS',/,9(1X,D12.5))
81 FORMAT (9(1X,D12.5))
82 FORMAT (2X,/)
83 FORMAT (///,1X,'THE P(0/-1) MATRIX IS',/)
END
SUBROUTINE ADD (A,B,N,M,C)
THIS SUBROUTINE ADDS THE NXM MATRICES A AND B, STORING THE
RESULT IN C
REAL*8 A,B,C
DIMENSION A(9,9),B(9,9),C(9,9)

DO 1 I=1,N
DO 1 J=1,M
1 C(I,J) = A(I,J)+B(I,J)
RETURN
END
SUBROUTINE A1 (A-H,O-Z)
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 MOK,M1K,M2K,MUK,MUKS
COMMON GK(9,9),S2K(9,9),S3K(9,9),PHI(9,9),C(9,9),H(9,9),EID(9,9),
1S1K(9,9),A2K(9,9),GK(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),
2A1K(9,9),S2K(9,9),S3K(9,9),RKM1(9,9),RKM2(9,9),S3KM1(9,9),
3S1KM1(9,9),RK(9,9),SOKM2(9,9),XK(9,9),VTMP1(9,9),XHIT(9,9),VTMP2(9,9),
4CTMP1(9,9),TMP2(9,9),PK(9,9),XK(9,9),XKM2(9,9),DK(9,9),N,L,NA,K,ND,NSAM,
5CGN(9,9),VTMP3(9,9),MUK(9,9),XKM1(9,9),XK(9,9,60),QK(9,9,60),
6IP,IPI,IH,IQ,PHK(9,9,60),HK(9,9,60),PKS2(9,9,60),
7PMUKS2(9,60),PKS1(9,9,60)
THIS SUBROUTINE COMPUTES THE CURRENT VALUE OF THE
NXN MATRIX A1(K) FROM EQ. (4.33)

CALL TRANS (S1K,N,NA,TMP1)
CALL PROD (RKM1,TMP1,NA,NA,N,TMP2)
CALL PROD (SOKM1,TMP2,N,NA,N,TMP1)
CALL PROD (S3K,TMP1,N,N,N,A1K)
RETURN

```

```

EVAL0415
EVAL0416
EVAL0417
EVAL0418
EVAL0419
EVAL0420
EVAL0421
EVAL0422
EVAL0423
EVAL0424
EVAL0425
EVAL0426
EVAL0427
EVAL0428
EVAL0429
EVAL0430
EVAL0431
EVAL0432
EVAL0434
EVAL0435
EVAL0436
EVAL0437
EVAL0438
EVAL0439
EVAL0440
EVAL0441
EVAL0442
EVAL0443
EVAL0444

```

```

EVAL0451
EVAL0452
EVAL0453
EVAL0454
EVAL0455
EVAL0456
EVAL0457
EVAL0458
EVAL0459

```

C C C C C

C C C C

EVAL0460
EVAL0461
EVAL0462
EVAL0463

```
END  
SUBROUTINE A2  
IMPLICIT REAL*8 (A-H,O-Z)  
COMMON MOK,M1K,M2K,MUK,MUKS  
COMMON GKS(9,9,60),PKS(9,9,60),PHI(9,9),C(9,9),H(9,9),EID(9,9),  
1S1K(9,9),S2K(9,9),S3K(9,9),GK(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),  
2A1K(9,9),A2K(9,9),RK(9,9),SOKM2(9,9),PK(9,9),XK(9,9),VTMP1(9,9),XKIT(9,9),  
3S1KM1(9,9),TMP2(9,9),TMP3(9,9),MUK(9,9),XKM1(9,9),XKM2(9,9),DK(9,9),N,L,NA,K,ND,NSAM,  
4TMP1(9,9),VTMP3(9,9),MUK(9,9),XKM1(9,9),XKM2(9,9),QK(9,9,60),PMUKS1(9,60),  
5CON(9,9),VTMP3(9,9),MUK(9,9),XKM1(9,9),XKM2(9,9),QK(9,9,60),PKS2(9,9,60)  
6IP,IP1,IH,IQ,PHK(9,9,60),PKS1(9,9,60),PKS2(9,9,60)  
7PMUKS2(9,60)  
THIS SUBROUTINE COMPUTES THE CURRENT VALUE OF THE  
NXN MATRIX A2(K)  
CALL TRANS (S2K,N,NA,TMP1)  
CALL PROD (RKM2,TMP1,NA,NA,N,TMP2)  
CALL PROD (S3K,M1,SOKM2,N,N,NA,TMP1)  
CALL ADD (TMP1,S1KM1,N,NA,A2K)  
CALL PROD (A2K,TMP2,N,NA,N,TMP1)  
CALL PROD (S3K,TMP1,N,N,A2K)  
RETURN  
END
```

EVAL0470
EVAL0471
EVAL0472
EVAL0473
EVAL0474
EVAL0475
EVAL0476
EVAL0477
EVAL0478
EVAL0479
EVAL0480
EVAL0481
EVAL0482

```
SUBROUTINE COVE  
IMPLICIT REAL*8 (A-H,O-Z)  
COMMON MOK,M1K,M2K,MUK,MUKS  
COMMON GKS(9,9,60),PKS(9,9,60),PHI(9,9),C(9,9),H(9,9),EID(9,9),  
1S1K(9,9),S2K(9,9),S3K(9,9),GK(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),  
2A1K(9,9),A2K(9,9),RK(9,9),SOKM2(9,9),PK(9,9),XK(9,9),VTMP1(9,9),XKIT(9,9),  
3S1KM1(9,9),TMP2(9,9),TMP3(9,9),MUK(9,9),XKM1(9,9),XKM2(9,9),DK(9,9),N,L,NA,K,ND,NSAM,  
4TMP1(9,9),VTMP3(9,9),MUK(9,9),XKM1(9,9),XKM2(9,9),QK(9,9,60),PMUKS1(9,60),  
5CON(9,9),VTMP3(9,9),MUK(9,9),XKM1(9,9),XKM2(9,9),QK(9,9,60),PKS2(9,9,60)  
6IP,IP1,IH,IQ,PHK(9,9,60),PKS1(9,9,60),PKS2(9,9,60)  
7PMUKS2(9,60)  
THIS SUBROUTINE CALCULATES THE COVARIANCE OF ESTIMATION  
ERROR USING EQ. (4.33)
```

EVAL0489
EVAL0490
EVAL0491
EVAL0492
EVAL0493
EVAL0494
EVAL0495
EVAL0496
EVAL0497
EVAL0498
EVAL0499
EVAL0500
EVAL0501
EVAL0502
EVAL0503

```
CALL ADD (A1K,A2K,N,N,A1K)  
CALL TRANS (A1K,N,N,A2K)  
CALL ADD (A1K,A2K,N,N,A1K)  
A1K NOW CONTAINS A1K+A2K + TRANSPOSE(A1K+A2K)  
CALL TRANS (S2K,N,NA,TMP1)  
CALL PROD (RKM2,TMP1,NA,NA,N,TMP2)  
CALL PROD (S2K,TMP2,N,NA,N,A2K)  
CALL ADD (A2K,A1K,N,N,A2K)  
CALL TRANS (S1K,N,NA,TMP1)  
CALL PROD (RKM1,TMP1,NA,NA,N,TMP2)  
CALL PROD (S1K,TMP2,N,NA,N,TMP1)  
CALL ADD (A2K,TMP1,N,N,A2K)
```

C C

C C C

C

EVAL0504
EVAL0505
EVAL0506
EVAL0507
EVAL0508
EVAL0509
EVAL0510
EVAL0511
EVAL0512
EVAL0513
EVAL0514
EVAL0515
EVAL0516

EVAL0523
EVAL0524
EVAL0525

EVAL0526
EVAL0527
EVAL0528
EVAL0529
EVAL0530
EVAL0531
EVAL0532
EVAL0533
EVAL0534
EVAL0535

EVAL0536
EVAL0537
EVAL0538
EVAL0539

```

CALL TRANS (SOK,N,NA,TMPI)
CALL PROD (RK,TMPI,NA,NA,N,TMP2)
CALL PROD (SOK,TMP2,N,NA,N,TMPI)
CALL ADD (A2K,TMPI,N,NA,A2K)
CALL TRANS (S3K,N,TMPI)
CALL PROD (PK,TMPI,N,TMP2)
CALL PROD (S3K,TMP2,N,N,TMPI)
CALL ADD (A2K,TMPI,N,N,PK)
RETURN
END
SUBROUTINE D
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 MOK,M1K,M2K,MUK,MUKS
COMMON GKS(9,9,60),PKS(9,9,60),PHI(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),
1S1K(9,9),S2K(9,9),S3K(9,9),GK(9,9),MOK(9,9),RK(9,9),RKM2(9,9),S3KM1(9,9),
2A1K(9,9),A2K(9,9),RK(9,9),SOKM2(9,9),PK(9,9),XK(9,9),VTMPI(9,9),
3S1KM1(9,9),TMP2(9,9),MUK(9,9),XKM1(9,9),XKM2(9,9),DK(9,9),NA,K,ND,NSAM,
4TMP1(9,9),VTMP3(9,9),PHK(9,9,60),HK(9,9,60),PKS2(9,9,60)
5CON(9,9),VTMP1(9,9),VTMP2(9,9),VTMP3(9,9),VTMP4(9,9),VTMP5(9,9),
6IP,IP1,IP2,IP3,IP4,IP5,IP6,IP7,IP8,IP9,IP10,IP11,IP12,IP13,IP14,IP15,IP16,
7PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)
THIS SUBROUTINE COMPUTES THE CURRENT VALUE OF THE N-VECTOR
D(K) FROM EQ.(4.10)

```

C C C

```

IF(IH.NE.0) GO TO 1
DO 1 I=1,NA

```

C

```

DO 1 J=1,N
H(I,J)=HK(I,J,K)
1 CONTINUE

```

C

```

CALL VPROD (H,XKM2,NA,N,VTMPI)
CALL VPROD (S2K,VTMPI,N,NA,VTMPI)
CALL VPROD (H,XKM1,NA,N,DK)
CALL VPROD (S1K,DK,N,NA,DK)
CALL VADD (DK,VTMPI,N,DK)
CALL PROD (SOK,H,N,NA,N,TMP1)
CALL SUB (TMP1,EID,N,N,TMP2)
CALL VPROD (TMP2,XK,N,N,VTMPI)
CALL VADD (VTMPI,DK,N,DK)
IF(IP.NE.0) GO TO 2

```

```

CALL UK
CALL VADD (DK,VTMP2,N,DK)
CONTINUE
RETURN

```

2

```

SUBROUTINE GAIN
IMPLICIT REAL*8 (A-H,O-Z)

```


EVAL0540

EVAL0547
EVAL0548
EVAL0549
EVAL0550

EVAL0551
EVAL0552
EVAL0553
EVAL0554
EVAL0555
EVAL0556
EVAL0557
EVAL0558
EVAL0559
EVAL0560
EVAL0561
EVAL0562
EVAL0563
EVAL0564

```

REAL*8 MOK,M1K,M2K,MUK,MUKS
COMMON GK{9,9,60},PKS{9,9,60},XKS(9,60),MUKS(9,60),SOK(9,9),
1S1K(9,9),S2K(9,9),S3K(9,9),PHI(9,9),C(9,9),H(9,9),EID(9,9),
2A1K(9,9),A2K(9,9),GK(9,9),MOK(9,9),M2K(9,9),SOKM1(9,9),
3S1KM1(9,9),RK(9,9),SOKM2(9,9),RKM1(9,9),RKM2(9,9),S3KM1(9,9),
4TMP1(9,9),TMP2(9,9),PK(9,9),XK(9,9),VTMP1(9,9),VTMP2(9,9),
5CON(9,9),VTMP3(9,9),MUK(9,9),XKM1(9,9),XKM2(9,9),DK(9,9),N,L,NA,K,ND,NSAM,
6IP,IPI,IH,IQ,PHK(9,9,60),HK(9,9,60),QK(9,9,60),PNUKSI(9,60),
7PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)

```

G(K) = P(K/K-1)*HT*(H*P(K/K-1)*HT + R)

DO 6 K=1,NSAM

IF(IH.NE.O) GO TO 7

DO 7 I=1,NA

DO 7 J=1,N

H(I,J)=HK(I,J,K)

7 CONTINUE

IF(IP.NE.O) GO TO 8

DO 8 I=1,N

DO 8 J=1,N

PHI(I,J)=PHK(I,J,K)

8 CONTINUE

IF(IQ.NE.O)GO TO 9

DO 9 I=1,N

DO 9 J=1,N

RKM2(I,J)=QK(I,J,K)

9 CONTINUE

CALL TRANS (H,NA,N,TMP2)

CALL PROD (PK,TMP2,N,N,NA,SOK)

CALL PROD (H,SOK,NA,N,NA,TMP1)

CALL ADD (TMP1,RKM1,NA,NA,TMP1)

IF (NA.EQ.1) GO TO 4

MD = ND

CALL GAUSS3 (NA,EPS,TMP1,TMP2,KER,MD)

CALL PROD (SOK,TMP2,N,NA,NA,GK)

NOTE HERE PKK(I,J) = P(K/K) WHERE

P(K/K) = (I-G(K)*H)*P(K/K-1)

1 CALL PROD (GK,H,N,NA,N,SOK)

CALL SUB (EID,SOK,N,N,TMP2)

CALL PROD (TMP2,PK,N,N,N,NA,S1K)

EVAL0565
EVAL0566
EVAL0567
EVAL0568
EVAL0569
EVAL0570
EVAL0571
EVAL0572
EVAL0573
EVAL0574
EVAL0575
EVAL0576
EVAL0577
EVAL0578
EVAL0579
EVAL0580
EVAL0581
EVAL0582
EVAL0583
EVAL0584
EVAL0585
EVAL0586
EVAL0587
EVAL0588
EVAL0589
EVAL0590
EVAL0591
EVAL0592
EVAL0593
EVAL0594

EVAL0601
EVAL0602
EVAL0603
EVAL0604
EVAL0605
EVAL0606
EVAL0607
EVAL0608
EVAL0609
EVAL0610

```

C      NOTE HERE      PKKM1(I,J) = P(K/K-1)  WHERE
C      P(K/K-1)=      PHI*P(K-1/K-1)*PHIT + Q
C      CALL TRANS      (PHI,N,N,TMP2)
C      CALL PROD      (SIK,TMP2,N,N,N,SOK)
C      CALL PROD      (PHI,SOK,N,N,N,TMP1)
C      CALL ADD      (TMP1,RKM2,N,N,N,PK)

C      DO 3 I=1,N
C
C      DO 2 J=1,NA
C      2 GKS(I,J,K) = GK(I,J)
C
C      DO 3 JK=1,N
C      3 PKS(I,JK,K) = SIK(I,JK)
C
C      GO TO 6
C
C      DO 5 I=1,N
C      5 GK(I,1) = SOK(I,1)/TMP1(1,1)
C
C      GO TO 1
C      6 CONTINUE
C
C      RETURN
C      END
C      SUBROUTINE LOADG      (A-H,Q-Z)
C      IMPLICIT REAL*8      (A-H,Q-Z)
C      REAL*8 MOK,M1K,M2K,MUK,MUKS
C      COMMON GKS(9,9,60),PKS(9,9,60),PHI(9,9),C(9,9),H(9,9),EID(9,9),SOK(9,9),
C      1SIK(9,9),S2K(9,9),S3K(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),
C      2AIK(9,9),A2K(9,9),GK(9,9),RKM2(9,9),RKM1(9,9),XK(9,9),VTMP1(9,9),
C      3SIKM1(9,9),RK(9,9),SOKM2(9,9),PK(9,9),XK(9,9),VTMP2(9,9),
C      4TMP1(9,9),TMP2(9,9),MUK(9,9),XKM1(9,9),XKM2(9,9),DK(9,9),NA,K,ND,NSAM,
C      5CON(9,9),VTMP3(9,9),PHK(9,9,60),HK(9,9,60),QK(9,9,60),PMUKS1(9,60),
C      6IP,IPI,IH,IQ,PKS1(9,9,60),PKS2(9,9,60)
C      7PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)
C
C      DO 1 I=1,N
C
C      DO 1 J=1,L
C      1 GK(I,J) = GKS(I,J,K)
C
C      RETURN
C      END
C      SUBROUTINE MEANE
C      THIS SUBROUTINE CALCULATES THE MEAN OF ESTIMATION ERROR

```


EVAL0611
EVAL0612
EVAL0613

```

FROM EQ. (4,34)
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 MOK,M1K,M2K,MUK,MUKS
COMMON GKS(9,9,60),PKS(9,9,60),XKS(9,60),MUKS(9,60),SOK(9,9),
1S1K(9,9),S2K(9,9),S3K(9,9),PHI(9,9),C(9,9),H(9,9),EID(9,9),
2A1K(9,9),A2K(9,9),GK(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),
3S1KM1(9,9),RK(9,9),SOKM2(9,9),RKM1(9,9),RKM2(9,9),S3KM1(9,9),
4TMP1(9,9),TMP2(9,9),PK(9,9),XK(9,9),VIMP1(9,9),XHIT(9,9),VIMP2(9,9),
5CON(9,9),VIMP3(9,9),MUK(9,9),XKM1(9,9),XKM2(9,9),DK(9,9),NA,K,ND,NSAM,
6IP,IPI,IH,IQ,PHK(9,9,60),HK(9,9,60),QK(9,9,60),PMUKS1(9,60),
7PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)
CALL VADD (MUK,XKM1,N,MUK)
CALL VPROD (S3K,MUK,N,N,MUK)
CALL VADD (MUK,DK,N,MUK)
RETURN
END

```

EVAL0620
EVAL0621
EVAL0622
EVAL0623
EVAL0624
EVAL0625
EVAL0626
EVAL0627
EVAL0628
EVAL0629

```

SUBROUTINE PRCD (A,B,N,M,L,C)
THIS SUBROUTINE COMPUTES THE MATRIX PRODUCT AB AND STORES THE
RESULT IN C
A = NXM, B = MXL, C = NXL
REAL*8 A,B,C,T
DIMENSION A(9,9),B(9,9),C(9,9),T(9,9)

```

EVAL0631
EVAL0632
EVAL0633
EVAL0634
EVAL0635
EVAL0636
EVAL0637
EVAL0638
EVAL0639
EVAL0640
EVAL0641
EVAL0642
EVAL0643
EVAL0644
EVAL0645
EVAL0646
EVAL0647
EVAL0648
EVAL0649
EVAL0650
EVAL0651
EVAL0652
EVAL0653
EVAL0654
EVAL0655

```

DO 1 I=1,N
DO 1 J=1,L
1 T(I,J) = 0.

DO 2 I=1,N
DO 2 J=1,L
DO 2 K=1,M
2 T(I,J) = T(I,J)+A(I,K)*B(K,J)

DO 3 I=1,N
DO 3 J=1,L
3 C(I,J) = T(I,J)

RETURN
END
SUBROUTINE RNOISE
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 MOK,M1K,M2K,MUK,MUKS
COMMON GKS(9,9,60),PKS(9,9,60),XKS(9,60),MUKS(9,60),SOK(9,9),

```



```

1 S1K(9,9), S2K(9,9), S3K(9,9), PHI(9,9), C(9,9), H(9,9), EID(9,9), SOKM1(9,9),
2 A1K(9,9), A2K(9,9), GK(9,9), MOK(9,9), M1K(9,9), M2K(9,9), S3KM1(9,9),
3 S1KM1(9,9), RK(9,9), SOKM2(9,9), RKM1(9,9), RKM2(9,9), S3KM1(9,9),
4 TMP1(9,9), TMP2(9,9), PK(9,9), XK(9,9), VTMP1(9,9), XHIT(9,9), VTMP2(9,9),
5 CON(9,9), VTMP3(9,9), MUK(9,9), XKM1(9,9), XKM2(9,9), DK(9,9), N, L, NA, K, ND, NSAM,
6 IP, IPI, IH, IQ, PHK(9,9,60), HK(9,9,60), QK(9,9,60), PMUKS1(9,60),
7 PMUKS2(9,60), PKS1(9,9,60), PKS2(9,9,60)
  RANGE = DSQRT(XKS(1,K)**2+XKS(3,K)**2)
  BEAR = DATAN2(XKS(3,K), XKS(1,K))
  VARR = CON(1)**2
  VARB = CON(2)**2
  ALPHA = (RANGE*CON(2))**2
  SIBR = DSIN(BEAR)
  COBR = DCOS(BEAR)
  RK(1,1) = ALPHA*SIBR**2+VARR*COBR**2
  RK(1,2) = (VARR-ALPHA)*COBR*SIBR
  RK(2,1) = RK(1,2)
  RK(2,2) = ALPHA*COBR**2+VARR*SIBR**2
  RETURN
END
SUBROUTINE STOR (A-H,O-Z)
  IMPLICIT REAL*8 (A-H,O-Z)
  REAL*8 MOK,M1K,M2K,MUK,MUKS
  COMMON GKS(9,9,60),PKS(9,9,60),PHI(9,9),C(9,9),H(9,9),EID(9,9),SOKM1(9,9),
1 S1K(9,9),S2K(9,9),S3K(9,9),SOKM2(9,9),MOK(9,9),M1K(9,9),M2K(9,9),S3KM1(9,9),
2 A1K(9,9),A2K(9,9),GK(9,9),RK(9,9),RKM1(9,9),RKM2(9,9),S3KM1(9,9),
3 S1KM1(9,9),RK(9,9),PK(9,9),XK(9,9),VTMP1(9,9),XHIT(9,9),VTMP2(9,9),
4 TMP1(9,9),TMP2(9,9),PK(9,9),XK(9,9),XKM1(9,9),XKM2(9,9),DK(9,9),N,L,NA,K,ND,NSAM,
5 CON(9,9),VTMP3(9,9),MUK(9,9),XKM1(9,9),XKM2(9,9),QK(9,9,60),PMUKS1(9,60),
6 IP,IPI,IH,IQ,PHK(9,9,60),HK(9,9,60),PKS1(9,9,60),PKS2(9,9,60)
7 PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)
  THIS SUBROUTINE STORES P(K),MU(K) AFTER THEY HAVE BEEN
  COMPUTED

```

CCCCC

```

DO I I=1,N
  MUKS(I,K) = MUK(I)

```

C

```

DO I J=1,N
  I PKS(I,J,K) = PK(I,J)

```

C

```

  RETURN
END

```

```

SUBROUTINE SUB (A,B,N,M,C)
THIS SUBROUTINE SUBTRACTS THE NXM MATRIX B FROM THE NXM MATRIX
A AND STORES THE RESULT IN C
REAL*8 A,B,C

```

CCCC

EVAL0662
EVAL0663
EVAL0664
EVAL0665
EVAL0666
EVAL0667
EVAL0668
EVAL0669
EVAL0670
EVAL0671
EVAL0672
EVAL0673
EVAL0674
EVAL0675
EVAL0676
EVAL0677

EVAL0684
EVAL0685
EVAL0686
EVAL0687
EVAL0688
EVAL0689
EVAL0690
EVAL0691
EVAL0692
EVAL0693
EVAL0694
EVAL0695
EVAL0696
EVAL0697
EVAL0698
EVAL0699
EVAL0700

DIMENSION A(9,9),B(9,9),C(9,9)

DO 1 I=1,N

DO 1 J=1,M

1 C(I,J) = A(I,J)-B(I,J)

RETURN

END

SUBROUTINE S3

IMPLICIT REAL*8 (A-H,O-Z)

REAL*8 MOK,M1K,M2K,MUK,MUKS

COMMON GKS(9,9),S3K(9,9),PKS(9,9),PHI(9,9),H(9,9),EID(9,9),

1S1K(9,9),A2K(9,9),GK(9,9),MOK(9,9),XKS(9,60),MUKS(9,60),SOK(9,9),

2A1K(9,9),A2K(9,9),S3K(9,9),GK(9,9),MOK(9,9),XKS(9,60),MUKS(9,60),SOK(9,9),

3S1K(9,9),A2K(9,9),S3K(9,9),GK(9,9),MOK(9,9),XKS(9,60),MUKS(9,60),SOK(9,9),

4TMP1(9,9),TMP2(9,9),PK(9,9),XK(9,9),VTMP1(9,9),XHIT(9,9),VTMP2(9,9),

5CON(9,9),VTMP3(9,9),MUK(9,9),XKM1(9,9),XKM2(9,9),DK(9,9),N,L,NA,K,ND,NSAM,

6IP,IP1,IH,IQ,PHK(9,9,60),HK(9,9,60),QK(9,9,60),PMUKS1(9,60),

7PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)

IF(IP.NE.0) GO TO 1

DO 1 I=1,N

DO 1 J=1,N

PHI(I,J)=PHK(I,J,K-1)

1 CONTINUE

IF(IH.NE.0) GO TO 2

DO 2 I=1,NA

DO 2 J=1,N

C(I,J)=HK(I,J,K)

2 CONTINUE

CALL PROD (GK,C,N,L,N,TMP1)

CALL SUB (EID,TMP1,N,N,TMP1)

CALL PROD (TMP1,PHI,N,N,N,S3K)

RETURN

END

SUBROUTINE TRACK

IF TRACK IS TO BE GENERATED ON-LINE IT IS DONE IN THIS SUBROUTINE

IMPLICIT REAL*8 (A-H,O-Z)

REAL*8 MOK,M1K,M2K,MUK,MUKS

COMMON GKS(9,9,60),PKS(9,9,60),XKS(9,60),MUKS(9,60),SOK(9,9),

1S1K(9,9),S2K(9,9),S3K(9,9),PHI(9,9),C(9,9),H(9,9),EID(9,9),

2A1K(9,9),A2K(9,9),GK(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),

EVAL0702
EVAL0703
EVAL0704
EVAL0705
EVAL0706
EVAL0707
EVAL0708
EVAL0709
EVAL0710
EVAL0711
EVAL0712

EVAL0719
EVAL0720
EVAL0721
EVAL0722
EVAL0723
EVAL0724
EVAL0725
EVAL0726


```

3 S1KM1(9,9),RK(9,9),SOKM2(9,9),RKM1(9,9),RKM2(9,9),S3KM1(9,9),
4 TMP1(9,9),TMP2(9,9),PK(9,9),XK(9,9),VTMP1(9,9),XHIT(9,9),VTMP2(9,9),
5 CON(9,9),VTMP3(9,9),MUK(9,9),XKM1(9,9),XKM2(9,9),DK(9,9),N,L,NA,K,ND,NSAM,
6 IP,IPI,IH,IQ,PHK(9,9,60),HK(9,9,60),QK(9,9,60),PMUKS1(9,60),
7 PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)

```

```

XKS(1,1)=2992.59D0
XKS(2,1)=-5295.85D0
XKS(3,1)=4.504D0
XKS(4,1)=5.06145D0
XKS(5,1)=500.D0

```

```
DO 3 K=2,NSAM
```

```
DO 1 I=1,N
  TMP1(I,1)=XKS(I,K-1)
1 CONTINUE

```

```
CALL PROD(PHI,TMP1,N,N,1,TMP1)
DO 2 I=1,N
  XKS(I,K)=TMP1(I,1)
2 CONTINUE

```

```
3 CONTINUE
RETURN
END

```

```

SUBROUTINE TRANS(A,N,M,C)
THIS SUBROUTINE FORMS THE MATRIX TRANSPOSE OF A STORING THE
RESULT IN C
A = NXM, C = MXN
REAL*8 A,C
DIMENSION A(9,9),C(9,9)

```

```
DO 1 I=1,N
```

```
DO 1 J=1,M
  C(J,I) = A(I,J)
1 CONTINUE

```

```
RETURN
END

```

```

SUBROUTINE VADD(X,Y,N,Z)
THIS SUBROUTINE COMPUTES THE SUM OF THE N-VECTORS X AND
Y AND STORES THE RESULT IN THE N-VECTOR Z

```

```
REAL*8 X(9),Y(9),Z(9)
```

```
DO 1 I=1,N
  Z(I) = X(I)+Y(I)
1 CONTINUE

```

```

EVAL0746
EVAL0747
EVAL0748
EVAL0749
EVAL0750

```

```

EVAL0752
EVAL0753
EVAL0754
EVAL0755
EVAL0756
EVAL0757
EVAL0758
EVAL0759
EVAL0760
EVAL0761
EVAL0762
EVAL0763

```

```

EVAL0765
EVAL0766
EVAL0767

```


EVAL0768
EVAL0769
EVAL0770
EVAL0771
EVAL0772
EVAL0773
EVAL0774
EVAL0775
EVAL0776

EVAL0778
EVAL0779
EVAL0780
EVAL0781
EVAL0782
EVAL0783
EVAL0784
EVAL0785
EVAL0786
EVAL0787
EVAL0788
EVAL0789
EVAL0790
EVAL0791
EVAL0792
EVAL0793
EVAL0794

EVAL0796
EVAL0797
EVAL0798
EVAL0799
EVAL0800
EVAL0801

```

C      RETURN
C      END
C      SUBROUTINE VPROD (A,X,M,N,Y)
C      THIS SUBROUTINE COMPUTES THE PRODUCT OF THE MXN MATRIX
C      A AND THE N-VECTOR X AND STORES THE RESULT IN THE
C      M-VECTOR Y
C
C      REAL*8 A(9,9),X(9),Y(9),T(9)
C
C      DO 1 I=1,M
C      T(I) = 0. DO
C
C      DO 1 J=1,N
C      1 T(I) = T(I)+A(I,J)*X(J)
C
C      DO 2 I=1,M
C      2 Y(I) = T(I)
C
C      RETURN
C      END
C      SUBROUTINE VSUB (X,Y,N,Z)
C      THIS SUBROUTINE COMPUTES THE DIFFERENCE X-Y OF THE TWO
C      N-VECTORS X & Y AND STORES THE RESULT IN THE N-VECTOR Z
C
C      REAL*8 X(9),Y(9),Z(9)
C
C      DO 1 I=1,N
C      1 Z(I) = X(I)-Y(I)
C
C      RETURN
C      END
C      SUBROUTINE OPLOT (MUKS,PKS,GKS,NSAM,N,NA)
C      THIS SUBROUTINE PLOTS THE OUTPUT
C
C      REAL*8 MUKS,PKS,GKS
C      DIMENSION MUKS(9,60),PKS(9,9,60),GKS(9,9,60),YP(60)
C
C      DO 88 K=1,NSAM
C      88 XP(K)=K
C
C      DO 90 I=1,N

```



```

THIS SUBROUTINE COMPUTES LINEARIZED MEASUREMENT MATRIX H(K)=HK(I,J,K)
C
C
C
A=-1.DO
VP=1640.DO
DO 1 K=1,NSAM
R=(XKS(1,K)**2)+(XKS(2,K)**2)
R=DSQRT(R)
F=VP+(XKS(3,K)*XKS(2,K))/R
F=(XKS(5,K)*VP)/F
HK(1,1,K)=-((A*XKS(2,K))/(R**2)
HK(1,2,K)=(A*XKS(1,K))/(R**2)
HK(1,3,K)=0.DO
HK(1,4,K)=1.DO
HK(1,5,K)=0.DO
C
HK(2,1,K)=(F**2)*XKS(3,K)*XKS(1,K)*XKS(2,K)
HK(2,1,K)=HK(2,1,K)/(XKS(5,K)*VP*(R**3))
HK(2,2,K)=-((F**2)*XKS(3,K)*(XKS(1,K)**2))
HK(2,2,K)=HK(2,2,K)/((R**3)*VP*XKS(5,K))
HK(2,3,K)=-((F**2)*XKS(2,K))
HK(2,3,K)=HK(2,3,K)/(XKS(5,K)*VP*R)
HK(2,4,K)=0.DO
HK(2,5,K)=F/XKS(5,K)
C
1 CONTINUE
RETURN
END
SUBROUTINE AK
C
C
C
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 MOK,M1K,M2K,MUK,MUKS
COMMON GKS(9,9,60),PKS(9,9,60),XKS(9,60),MUKS(9,60),SOK(9,9),
1S1K(9,9),S2K(9,9),S3K(9,9),PHI(9,9),C(9,9),H(9,9),EID(9,9),
2A1K(9,9),A2K(9,9),GK(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),
3S1KM1(9,9),RK(9,9),SOKM2(9,9),RKM1(9,9),RKM2(9,9),S3KM1(9,9),
4TMP1(9,9),TMP2(9,9),PK(9,9),XK(9,9),VTMP1(9,9),XHI(9,9),VTMP2(9,9),
5CON(9,9),VTMP3(9,9),MUK(9,9),XKM1(9,9),XKM2(9,9),DK(9,9),N,L,NA,K,ND,NSAM,
6IP,IPI,IH,IQ,PIK(9,9,60),HK(9,9,60),QK(9,9,60),PMUKSI(9,60),
7PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)
THIS SUBROUTINE COMPUTES LINEARIZED STATE MATRIXPH(K)=PHK(I,J,K)
C
C
C

```


EVAL0727
EVAL0728

```

C DO 1 K=1, NSAM
C 1 CONTINUE
C RETURN
C END
C SUBROUTINE UK
C IMPLICIT REAL*8 (A-H,O-Z)
C REAL*8 MOK,M1K,M2K,MUK,MUKS
C COMMON GKS(9,9,60),PKS(9,9,60),XKS(9,60),MUKS(9,60),SOK(9,9),
1 S1K(9,9),S2K(9,9),S3K(9,9),PHI(9,9),C(9,9),H(9,9),EID(9,9),
2 A1K(9,9),A2K(9,9),GK(9,9),MOK(9,9),M1K(9,9),M2K(9,9),SOKM1(9,9),
3 S1KM1(9,9),RK(9,9),SOKM2(9,9),RKM1(9,9),RKM2(9,9),S3KM1(9,9),
4 TMP1(9,9),TMP2(9,9),PK(9,9),XK(9,9),VTMP1(9,9),XHI(9,9),VTMP2(9,9),
5 CON(9,9),VTMP3(9,9),MUK(9,9),XKM1(9,9),XKM2(9,9),DK(9,9),N,L,NA,K,ND,NSAM,
6 IP1,IP2,IP3,IP4,IP5,IP6,IP7,IP8,IP9,IP10,IP11,IP12,IP13,IP14,IP15,IP16,IP17,IP18,IP19,IP20,
7 PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)
C THIS SUBROUTINE COMPUTES THE FORCING INPUT U(K) WHICH RESULTS FROM
C LINERIZATION U(K)=A(X(K),K)-PHK(I,J,K)X(K)
C DO 1 I=1,N
C VTMP1(I)=XKS(I,K-1)
C DO 1 J=1,N
C 1 TMP1(I,J)=PHK(I,J,K-1)
C CALL VPROD (TMP1,VTMP1,N,N,VTMP2)
C DO 2 I=1,N
C 2 VTMP1(I)=XKS(I,K)
C CALL VSUB (VTMP1,VTMP2,N,N,VTMP2)
C CALCULATE (I-G(K)C(K))U'(K-1)
C CALL PROD (GK,C,N,L,N,TMP1)
C CALL SUB (EID,TMP1,N,N,TMP1)
C CALL VPROD (TMP1,VTMP2,N,N,VTMP2)
C RETURN
C END
C SUBROUTINE PROB
C IMPLICIT REAL*8 (A-H,O-Z)
C REAL*8 MOK,M1K,M2K,MUK,MUKS
C COMMON GKS(9,9,60),PKS(9,9,60),XKS(9,60),MUKS(9,60),SOK(9,9),
1 S1K(9,9),S2K(9,9),S3K(9,9),PHI(9,9),C(9,9),H(9,9),EID(9,9),

```



```

C DO 6 J=1,N
C TMP2(I,J)=PKS2(I,J,K)
C CALL ADD(TMP2,TMP1,N,N,TMP1)
C DO 7 I=1,N
C DO 7 J=1,N
C S2K(I,J)=TMP1(I,J)*(1.-P)
C DO 8 I=1,N
C DO 8 J=1,N
C PKS(I,J,K)=SIK(I,J)+S2K(I,J)
C 9 CONTINUE
C RETURN
C END
C SUBROUTINE QON
C IMPLICIT REAL*8 (A-H,O-Z)
C REAL*8 MOK,M1K,M2K,MUK,MUKS
C COMMON GKS(9,9,60),PKS(9,9,60),PHI(9,9),C(9,9),H(9,9),EID(9,9),SOK(9,9),
C 1SIK(9,9),S2K(9,9),S3K(9,9),MOK(9,9),GK(9,9),SOKM2(9,9),RKM1(9,9),XK(9,9),VTMP1(9,9),XHIT(9,9),
C 2AIK(9,9),A2K(9,9),S3K(9,9),MOK(9,9),GK(9,9),SOKM2(9,9),RKM1(9,9),XK(9,9),VTMP1(9,9),XHIT(9,9),
C 3SIKMI(9,9),TMP2(9,9),PK(9,9),XKMI(9,9),XKM2(9,9),DK(9,9),NA,K,ND,NSAM,
C 4TMP1(9,9),TMP2(9,9),PK(9,9),XK(9,9),XKM1(9,9),XKM2(9,9),DK(9,9),NA,K,ND,NSAM,
C 5CCN(9,9),VTMP3(9,9),MUK(9,9),HK(9,9,60),QK(9,9,60),PMUKS1(9,60),
C 6IP,IPL,IH,IQ,PHK(9,9,60),QK(9,9,60),QK(9,9,60),QK(9,9,60),QK(9,9,60),
C 7PMUKS2(9,60),PKS1(9,9,60),PKS2(9,9,60)
C DELT=100.D0
C SIGVS=31.7D-05
C SIGTS=3.04D-05
C SIGFS=.25D-06
C A=-1.D0
C DELT1=DELT**2
C DO 1 K=1,NSAM
C QK(1,1,K)=(XKS(2,K)**2)*DELT1*SIGTS
C QK(1,2,K)=-XKS(1,K)*XKS(2,K)*DELT1*SIGTS
C QK(1,3,K)=0.D0
C QK(1,4,K)=A*XKS(2,K)*DELT1*SIGTS
C QK(1,5,K)=0.D0
C QK(2,2,K)=((DELT1**2)/4.D0)*SIGVS+(XKS(1,K)**2)*DELT1*SIGTS
C QK(2,3,K)=((DELT**3)/2.D0)*SIGVS

```



```

      QK(2,4,K)=- (A*XKS(1,K)*DELT1*SIGTS)
      QK(2,5,K)=0.D0
      QK(3,3,K)=DELT1*SIGVS
      QK(3,4,K)=0.D0
      QK(3,5,K)=0.D0
      QK(4,4,K)=DELT1*SIGTS
      QK(4,5,K)=0.D0
      QK(5,5,K)=DELT1*SIGFS
      QK(2,1,K)=QK(1,2,K)
      QK(3,1,K)=QK(1,3,K)
      QK(3,2,K)=QK(2,3,K)
      QK(4,1,K)=QK(1,4,K)
      QK(4,2,K)=QK(2,4,K)
      QK(4,3,K)=QK(3,4,K)
      QK(5,1,K)=QK(1,5,K)
      QK(5,2,K)=QK(2,5,K)
      QK(5,3,K)=QK(3,5,K)
      QK(5,4,K)=QK(4,5,K)
      1 CONTINUE
      RETURN
      END

```


CCCCCCCC

THIS PROGRAM PERFORMS MONTE CARLO SIMULATION OF STATE ESTIMATORS
OF WHICH THE KALMAN FILTER IS ONE EXAMPLE. THERE ARE SEVERAL
OPTIONS AVAILABLE AS INDICATED IN THE DETAILED COMMENTS BELOW.
IT SHOULD BE NOTED THAT ALL COMPUTATIONS OF GAINS USING THE
SUBROUTINE GAIN ARE PERFORMED IN DOUBLE PRECISION. THUS ALL
ARRAYS FOR USE IN "GAIN" MUST BE PREPARED ACCORDINGLY.

```
REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,PKKM2,G,PKK,Q,EI
COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),
1TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),
2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),
3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),
4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VTMP(5),Z(5),V(5),SIGV(5),
5XHATZ(5),C(6),IEXT,IPHI,IH,
6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRD,IXZ,IV,IW,IEST,ND,M
DIMENSION XP(50),YP(60),PKKM2(5,5)
```

```
N=ORDER OF SYSTEM MODEL AND FILTER (DIMENSION OF X,XHAT)
M=NUMBER OF MEASUREMENTS (DIMENSION OF THE VECTOR Z)
IN=NUMBER OF INPUT RANDOM FORCING FCNS (=DIMENSION OF W)
NSAM=NUMBER OF TIME SAMPLES
NENS=NUMBER OF MEMBERS IN ENSEMBLE
READ (5,81) N,M,IN,NSAM,NENS
```

```
READ (5,82) ND
THE VALUE OF ND READ IN MUST EQUAL THE ROW (AND COLUMN) DIMENSION
SPECIFIED FOR THE SQUARE MATRIX "TEMP1", E.G. IF TEMP1(3,3) IS
SPECIFIED IN THE COMMON STATEMENT "ND" MUST BE EQUAL TO 3.
```

```
IG=-1 -- GAINS COMPUTED OFF-LINE AND READ IN
0 -- GAINS COMPUTED ONLY ONCE BEFORE STARTING MONTE CARLO
1 -- GAINS COMPUTED FOR EACH MEMBER OF ENSEMBLE
```

```
IFLR=0 -- R IS READ IN
IFLR.NE.0 -- R IS COMPUTED ON-LINE AT EACH TIME SAMPLE
IEST=0 -- STANDARD KALMAN FILTER EQUATION IS USED
IEST.NE.0 -- STD. KALMAN FILTER EQ. NOT USED
ITRK= 0 -- SEVERAL TRACKS GENERATED FROM STD LINEAR EQS
```

CCCCCCCCCCCCCCCC C C CCCCCCCCCCCCCCCCCCCCCC

MCSP0000
MCSP0001
MCSP0002
MCSP0003
MCSP0004
MCSP0005
MCSP0006
MCSP0007

MCSP0016
MCSP0017
MCSP0018
MCSP0019
MCSP0020
MCSP0021
MCSP0022
MCSP0023
MCSP0024
MCSP0025
MCSP0026
MCSP0027
MCSP0028
MCSP0029
MCSP0030
MCSP0031
MCSP0032
MCSP0033
MCSP0034
MCSP0035
MCSP0036
MCSP0037
MCSP0038
MCSP0039
MCSP0040
MCSP0041
MCSP0042
MCSP0043
MCSP0044
MCSP0045
MCSP0046
MCSP0047

MCSP00093
 MCSP00094
 MCSP00095
 MCSP00096
 MCSP00097
 MCSP00098
 MCSP00099
 MCSP0100
 MCSP0101
 MCSP0102
 MCSP0103
 MCSP0104
 MCSP0105
 MCSP0106
 MCSP0107
 MCSP0108
 MCSP0109
 MCSP0110
 MCSP0111
 MCSP0112
 MCSP0113
 MCSP0114
 MCSP0115
 MCSP0116
 MCSP0117
 MCSP0118
 MCSP0119
 MCSP0120
 MCSP0121
 MCSP0122
 MCSP0123
 MCSP0124
 MCSP0125
 MCSP0126
 MCSP0127
 MCSP0128
 MCSP0129
 MCSP0130
 MCSP0131
 MCSP0132
 MCSP0133
 MCSP0134
 MCSP0135
 MCSP0136
 MCSP0137
 MCSP0138
 MCSP0139
 MCSP0140

```

IGAM=0 -- GAMMA MATRIX TO BE READ IN (D.P.)
ISIGV=0 -- STD. DEVIATIONS OF MEASUREMENT NOISE TO BE READ IN (SP)
ISIGW=0 -- STD. DEVIATIONS OF RANDOM INPUTS TO BE READ IN (SP)
IXHZ=0 -- INITIAL VALUE OF THE PREDICTED VALUE XH(0/-1) TO
        BE READ IN (S.P.)

IC=1 -- ONE INITIAL CONDITION VALUE USED FOR THE STATE
        OTHERWISE THE VALUE OF X(0) IS GENERATED BY A RANDOM
        NUMBER GENERATOR CALLED BY "XZERO"

READ (5,86) IPHI,IH,IR,IPKKM1,IGAM,ISIGV,ISIGW,IXHZ,IC
CALL OVFLOW
IW = 6395217
IV = 1936748
IXZ = 135769
  
```

THE FOLLOWING SECTION PRINTS OUT A DESCRIPTION OF THE RUN AS
 SPECIFIED BY THE USER'S FLAGS

```

WRITE (6,156)
WRITE (6,87)
IF (IG.EQ.0) GO TO 1
IF (IG.EQ.1) GO TO 2
WRITE (6,88)
GO TO 3
WRITE (6,89)
1 GO TO 3
2 WRITE (6,90)
3 IF (IEST.EQ.0) GO TO 4
WRITE (6,91)
GO TO 5
4 WRITE (6,92)
5 IF (ITRK.EQ.0) GO TO 8
IF (ITRK.EQ.-1) GO TO 7
IF (ITRO.EQ.0) GO TO 6
WRITE (6,93)
GO TO 9
6 WRITE (6,94)
7 GO TO 9
7 WRITE (6,95)
8 GO TO 9
8 WRITE (6,96)
  
```

CCCCCCCCCCCCCCCC

CCCCCC


```

9 IF (ISTAT.EQ.0) GO TO 10
  WRITE (6,97)
  GO TO 11
10 WRITE (6,98)
11 IF (IQ.EQ.0) GO TO 13
  IF (IQ.EQ.1) GO TO 12
  WRITE (6,99)
  GO TO 14
12 WRITE (6,100)
  GO TO 14
13 WRITE (6,101)
14 IF (IFLR.EQ.0) GO TO 15
  WRITE (6,102)
  GO TO 16
15 WRITE (6,103)
16 WRITE (6,104)
  IF (IPHI.EQ.0) WRITE (6,105)
  IF (IPH.EQ.0) WRITE (6,106)
  IF (IFLR.EQ.0) WRITE (6,107)
  IF (IQ.EQ.0) GO TO 17
  IF (IQ.EQ.-1) GO TO 18
  WRITE (6,108)
  GO TO 18
17 WRITE (6,109)
18 IF (IGAM.EQ.0) WRITE (6,110)
  IF (ISIGV.EQ.0) WRITE (6,111)
  IF (ISIGW.EQ.0) WRITE (6,112)
  IF (IXHZ.EQ.0) WRITE (6,113)
  IF (IC.NE.1) WRITE (6,114)
  IF (IPKKM1.EQ.0) WRITE (6,115)

THE FOLLOWING SECTION PRINTS OUT A DESCRIPTION OF THE OUTPUT
DATA CALLED FJR

WRITE (6,116)
IF (IPRT.NE.0) GO TO 19
WRITE (6,117)
IF (IG.EQ.0) WRITE (6,118)
WRITE (6,119)
IF (ISTAT.NE.0) WRITE (6,120)
GO TO 20
19 WRITE (6,121)
20 IF (IPLT.NE.0) GO TO 21
  WRITE (6,122)
  IF (IGPLT.EQ.1) WRITE (6,123)
  IF (ITHVPL.EQ.1) WRITE (6,124)

```

CCCCC

MCSP0141
 MCSP0142
 MCSP0143
 MCSP0144
 MCSP0145
 MCSP0146
 MCSP0147
 MCSP0148
 MCSP0149
 MCSP0150
 MCSP0151
 MCSP0152
 MCSP0153
 MCSP0154
 MCSP0155
 MCSP0156
 MCSP0157
 MCSP0158
 MCSP0159
 MCSP0160
 MCSP0161
 MCSP0162
 MCSP0163
 MCSP0164
 MCSP0165
 MCSP0166
 MCSP0167
 MCSP0168
 MCSP0169
 MCSP0170
 MCSP0171
 MCSP0172
 MCSP0173
 MCSP0174
 MCSP0175
 MCSP0176
 MCSP0177
 MCSP0178
 MCSP0179
 MCSP0180
 MCSP0181
 MCSP0182
 MCSP0183
 MCSP0184
 MCSP0185
 MCSP0186
 MCSP0187
 MCSP0188

MCSP0189
 MCSP0190
 MCSP0191
 MCSP0192
 MCSP0193
 MCSP0194
 MCSP0195
 MCSP0196
 MCSP0197
 MCSP0198
 MCSP0199
 MCSP0200
 MCSP0201
 MCSP0202
 MCSP0203
 MCSP0204
 MCSP0205
 MCSP0206
 MCSP0207
 MCSP0208
 MCSP0209
 MCSP0210
 MCSP0211
 MCSP0212
 MCSP0213
 MCSP0214
 MCSP0215
 MCSP0216
 MCSP0217
 MCSP0218
 MCSP0219
 MCSP0220
 MCSP0221
 MCSP0222
 MCSP0223
 MCSP0224
 MCSP0225
 MCSP0226
 MCSP0227
 MCSP0228
 MCSP0229
 MCSP0230
 MCSP0231
 MCSP0232
 MCSP0233
 MCSP0234
 MCSP0235
 MCSP0236

IF (IMTPLT.EQ.1) WRITE (6,125)
 IF (ISMPLT.EQ.1) WRITE (6,126)
 IF (ISVPLT.EQ.1) WRITE (6,127)
 GO TO 22
 21 WRITE (6,128)
 22 CONTINUE
 WRITE (6,156)
 WRITE (6,129)
 WRITE (6,130) N,M,IN,NSAM,NENS,ND

THE FOLLOWING SECTION READS THE SPECIFIED INPUT MATRICES

IF (IPHI.NE.0) GO TO 24
 CALL MREAD (PHI,N,N)

DO 23 I=1,N

DO 23 J=1,N
 23 PHIS(I,J) = PHI(I,J)

WRITE (6,131)
 CALL MWWRITE (PHI,N,N)

24 IF (IH.NE.0) GO TO 26
 CALL MREAD (H,M,N)

DO 25 I=1,M

DO 25 J=1,N
 25 HS(I,J) = H(I,J)

WRITE (6,132)
 CALL MWWRITE (H,M,N)

26 IF (IR.NE.0) GO TO 27
 CALL MREAD (R,M,M)
 WRITE (6,133)
 CALL MWWRITE (R,M,M)

27 IF (IQ.NE.1) GO TO 28
 CALL MREAD (COVW,IN,IN)
 WRITE (6,134)
 CALL MWWRITE (COVW,IN,IN)
 GO TO 29

28 IF (IQ.NE.O) GO TO 29
 CALL MREAD (Q,N,N)
 WRITE (6,135)
 CALL MWRITE (Q,N,N)
 C
 C
 29 IF (IGAM.NE.O) GO TO 31
 CALL MREAD (GAMMA,N,IN)
 DO 30 I=1,N
 C
 DO 30 J=1,IN
 30 GAMMA(I,J) = GAMMA(I,J)
 C
 WRITE (6,136)
 CALL MWRITE (GAMMA,N,IN)
 C
 C
 31 IF (IPKKM1.NE.O) GO TO 32
 CALL MREAD (PKKM2,N,N)
 WRITE (6,137)
 CALL MWRITE (PKKM2,N,N)
 DO 165 I=1,N
 DO 165 J=1,N
 PKKM1(I,J)=PKKM2(I,J)
 165 CONTINUE
 C
 C
 32 IF (ISIGV.NE.O) GO TO 33
 CALL VREAD (SIGV,M)
 WRITE (6,138)
 CALL VWRITE (SIGV,M)
 C
 C
 33 IF (ISIGW.NE.O) GO TO 34
 CALL VREAD (SIGW,IN)
 WRITE (6,139)
 CALL VWRITE (SIGW,IN)
 C
 C
 34 IF (IXHZ.NE.O) GO TO 35
 CALL VREAD (XHATZ,N)
 WRITE (6,140)
 CALL VWRITE (XHATZ,N)
 C
 C
 35 IF (IC.EQ.1) GO TO 36
 IC.NE.1 MEANS THAT MEANS AND STD. DEVIATIONS OF THE INITIAL STATE
 C
 C

MCSP0237
 MCSP0238
 MCSP0239
 MCSP0240
 MCSP0241
 MCSP0242
 MCSP0243
 MCSP0244
 MCSP0245
 MCSP0246
 MCSP0247
 MCSP0248
 MCSP0249
 MCSP0250
 MCSP0251
 MCSP0252
 MCSP0253
 MCSP0254
 MCSP0255
 MCSP0256
 MCSP0257
 MCSP0258
 MCSP0259
 MCSP0260
 MCSP0261
 MCSP0262
 MCSP0263
 MCSP0264
 MCSP0265
 MCSP0266
 MCSP0267
 MCSP0268
 MCSP0269
 MCSP0270
 MCSP0271
 MCSP0272
 MCSP0273
 MCSP0274
 MCSP0275
 MCSP0276
 MCSP0277
 MCSP0278
 MCSP0279
 MCSP0280

MC SP0281
MC SP0282
MC SP0283
MC SP0284
MC SP0285
MC SP0286
MC SP0287
MC SP0288
MC SP0289
MC SP0290
MC SP0291
MC SP0292
MC SP0293
MC SP0294
MC SP0295
MC SP0296
MC SP0297
MC SP0298
MC SP0299
MC SP0300
MC SP0301
MC SP0302
MC SP0303
MC SP0304
MC SP0305
MC SP0306
MC SP0307
MC SP0308
MC SP0309
MC SP0310
MC SP0311
MC SP0312
MC SP0313
MC SP0314
MC SP0315
MC SP0316
MC SP0317
MC SP0318
MC SP0319
MC SP0320
MC SP0321
MC SP0322
MC SP0323
MC SP0324
MC SP0325
MC SP0326
MC SP0327
MC SP0328

```

C      VALUE MUST BE READ IN. OTHERWISE NOT READ IN.
C      CALL VREAD (XZMEAN,N)
C      WRITE (6,141)
C      CALL VWRITE (XZMEAN,N)

C
C      CALL VREAD (SIGXZ,N)
C      WRITE (6,142)
C      CALL VWRITE (SIGXZ,N)
C      GO TO 40

C
C      36 READ (5,144) (XS(I,1),I=1,N)
C      INITIAL CONDITION HAS BEEN READ
C      WRITE (6,143)
C      WRITE (6,146) (XS(I,1),I=1,N)
C      IF (ITRK.NE.1) GO TO 40
C      IF (ITRO.NE.0) GO TO 38

C
C      DO 37 K=2,NSAM
C      37 READ (5,144) (XS(I,K),I=1,N)

C
C      GO TO 39
C      38 CALL TRACK
C      IF TRACK CALLED HERE IT SHOULD BE WRITTEN TO GENERATE AND
C      STORE THE TRACK IN XS(N,K) FOR K=2(NOTE),NSAM

C
C      39 WRITE (6,145)
C      WRITE (6,146) (XS(I,1),I=1,N)
C      WRITE (6,146) (XS(I,NSAM),I=1,N)
C      40 CONTINUE

C
C      THE FOLLOWING SECTION PREPARES FOR THE MONTE CARLO LOOP
C      FORM NXN IDENTITY MATRIX IN DOUBLE PRECISION
C      DO 41 I=1,N
C      DO 41 J=1,N
C      41 EI(I,J) = 0.D0
C      IF (I.EQ.J) EI(I,J)=1.D0
C      GIVEN THE MATRIX GAMMA AND THE COVARIANCE OF W COMPUTE Q
C      USING DOUBLE PRECISION ARITHMETIC
C      IF (IQ.NE.1) GO TO 42
C      CALL QMAT
C      WRITE (6,135)

```


MCSP03329
MCSP03330
MCSP03331
MCSP03332
MCSP03333
MCSP03334
MCSP03335
MCSP03336
MCSP03337
MCSP03338
MCSP03339
MCSP03340
MCSP03341
MCSP03342
MCSP03343
MCSP03344
MCSP03345
MCSP03346
MCSP03347
MCSP03348
MCSP03349
MCSP03350
MCSP03351
MCSP03352
MCSP03353
MCSP03354
MCSP03355
MCSP03356
MCSP03357
MCSP03358
MCSP03359
MCSP03360
MCSP03361
MCSP03362
MCSP03363
MCSP03364
MCSP03365
MCSP03366
MCSP03367
MCSP03368
MCSP03369
MCSP03370
MCSP03371
MCSP03372
MCSP03373
MCSP03374
MCSP03375

```

      CALL MWRITE (Q,N,N)
42  IF (IG.NE.-1) GO TO 44
      DO 43 K=1,NSAM
      DO 43 I=1,N
43  READ (5,144) (GKS(I,J,K),J=1,M)
      GO TO 47
44  IF (IG.NE.0) GO TO 47
      DO 46 K=1,NSAM
      CALL GAIN
      DO 46 I=1,N
      DO 45 L=1,N
45  PKKS(I,L,K) = PKK(I,L)
      DO 46 J=1,M
46  GKS(I,J,K) = G(I,J)
47  CONTINUE
      IF GAINS WERE TO BE READ IN (IG=-1) OR COMPUTED ONLY
      ONCE(IG=0), THIS HAS NOW BEEN DONE

      SET UP ARRAYS FOR COMPUTING STATISTICS
      DO 48 K=1,NSAM
      DO 48 J=1,N
      XM(J,K) = 0.
      ERR(J,K) = 0.
      DO 48 L=1,N
48  VAR(J,L,K) = 0.

      BEGIN MAIN ITERATION LOOP HERE
      DO 54 ITER=1,NENS
      IF (IC.EQ.1) GO TO 49
      CALL XZERO
      DO 50 I=1,N
49  DO 50 I=1,N

```


50 XHKM1(I) = XHATZ(I)

C

DO 164 I=1,N
DO 164 J=1,N
PKM1(I,J)=PKM2(I,J)
164 CONTINUE

C

DO 54 K=1,NSAM
FORM NOISY MEASUREMENT FROM TRUE STATE VALUE

C

DO 51 I=1,N
51 X(I) = XS(I,K)

C

CALL MEAS
GAIN IS NOT TO BE COMPUTED ON-LINE IF IG.NE.1
IF (IG.NE.1) GO TO 53

C

Q IS TO BE COMPUTED ON-LINE IF IFLQ.NE.0
R IS TO BE COMPUTED ON-LINE IF IFLR.NE.0

C

IF (IFLR.NE.0) CALL RON
CALL GAIN

C

DO 52 I=1,N

C

DO 52 J=1,M
52 GKS(I,J,K) = G(I,J)

C

UPDATE THE STATE ESTIMATE

C

53 IF (IEXT.EQ.0) CALL ESTIM

C

UPDATE RUNNING SUMS USED IN COMPUTING STATISTICS
CALL STAT
IF (K.EQ.NSAM) GO TO 54

C

UPDATE TRACK BY COMPUTING X(K+1)
IF (ITRK.NE.1) CALL TRACK
54 CONTINUE

C

DIVIDE RUNNING SUMS COMPUTED BY SUBROUTINE STAT BY ENSEMBLE
SIZE TO COMPUTE STATISTICS
ENS = NENS

C

C

MCSP0378
MCSP0379
MCSP0380
MCSP0381
MCSP0382
MCSP0383
MCSP0384
MCSP0385
MCSP0386
MCSP0387
MCSP0388
MCSP0389
MCSP0390
MCSP0391
MCSP0393
MCSP0394
MCSP0395
MCSP0396
MCSP0397
MCSP0398
MCSP0399
MCSP0400
MCSP0401
MCSP0402
MCSP0403
MCSP0404
MCSP0405
MCSP0406
MCSP0407
MCSP0408
MCSP0409
MCSP0410
MCSP0411
MCSP0412
MCSP0413
MCSP0414
MCSP0415
MCSP0416
MCSP0417
MCSP0418
MCSP0419


```

C      DO 56 K=1,NSAM
C
C      DO 56 J=1,N
C      IF (ITRK.EQ.1) GO TO 55
C      XM(J,K) = XM(J,K)/ENS
C      55 ERR(J,K) = ERR(J,K)/ENS
C      56 VAR(J,J,K) = VAR(J,J,K)/ENS-ERR(J,K)**2
C
C      IF (ISTAT.EQ.0) GO TO 58
C
C      COMPUTE OFF-DIAGONAL TERMS IN COVARIANCE OF ESTIMATION
C      ERROR MATRIX IF ISTAT.NE.0
C
C      DO 57 K=1,NSAM
C
C      DO 57 L=2,N
C      LM1 = L-1
C
C      DO 57 J=1,LM1
C      57 VAR(L,J,K) = VAR(L,J,K)/ENS-ERR(L,K)*ERR(J,K)
C
C      58 CONTINUE
C
C      IF (IPRT.NE.0) GO TO 64
C      WRITE (6,147)
C      WRITE GAINS, THEORETICAL COVARIANCES OF ESTIMATION ERROR
C      IF ONE SET OF GAINS HAS BEEN USED.
C      WRITE (6,148)
C
C      DO 59 K=1,NSAM
C      WRITE (6,149) K
C
C      DO 59 I=1,N
C      59 WRITE (6,146) (GKS(I,J,K), J=1,M)
C
C      WRITE (6,150)
C
C      DO 60 K=1,NSAM
C      WRITE (6,151) K
C
C      DO 60 I=1,N
C      60 WRITE (6,146) (PKKS(I,J,K), J=1,N)
C
C      61 WRITE (6,156)

```


MCSP0469
MCSP0470
MCSP0471
MCSP0472
MCSP0473
MCSP0474
MCSP0475
MCSP0476
MCSP0477
MCSP0478
MCSP0479
MCSP0480
MCSP0481
MCSP0482
MCSP0483
MCSP0484
MCSP0485
MCSP0486
MCSP0487
MCSP0488
MCSP0489
MCSP0490
MCSP0491
MCSP0492
MCSP0493
MCSP0494
MCSP0495
MCSP0496
MCSP0497
MCSP0498
MCSP0499
MCSP0500
MCSP0501
MCSP0502
MCSP0503
MCSP0504
MCSP0505
MCSP0506
MCSP0507
MCSP0508
MCSP0509
MCSP0510
MCSP0511
MCSP0512
MCSP0513
MCSP0514
MCSP0515
MCSP0516

```

C      WRITE (6,152)
      WRITE (6,153)
C      DO 62 K=1,NSAM
      WRITE (6,155)
C
C      DO 62 I=1,N
      WRITE (6,154) K,I,XM(I,K),ERR(I,K),VAR(I,I,K)
C      62
      WRITE (6,156)
      IF (ISTAT.EQ.0) GO TO 64
      WRITE (6,157)
C
C      DO 63 K=1,NSAM
      WRITE (6,158) K
C
C      DO 63 I=1,N
      WRITE (6,146) (VAR(I,L,K),L=1,I)
C      63
      WRITE (6,156)
      IF (IPLT.NE.0) GO TO 80
C      64
      DO 65 K=1,NSAM
      XP(K) = K
C      65
      IF (IGPLT.NE.1) GO TO 68
C
C      DO 67 I=1,N
C      DO 67 J=1,M
C
C      DO 66 K=1,NSAM
      YP(K) = GKS(I,J,K)
C      66
      WRITE (6,156)
      CALL PLOTP (XP,YP,NSAM,0)
C      67
      WRITE (6,159) I,J
C      68
      IF (ITHVPL.NE.1) GO TO 71
C
C      DO 70 I=1,N
C
C      DO 69 K=1,NSAM
      YP(K) = PKKS(I,I,K)
C      69
      WRITE (6,156)
      CALL PLOTP (XP,YP,NSAM,0)
C      70
      WRITE (6,160) I,I

```


MC SP0517
MC SP0518
MC SP0519
MC SP0520
MC SP0521
MC SP0522
MC SP0523
MC SP0524
MC SP0525
MC SP0526
MC SP0527
MC SP0528
MC SP0529
MC SP0530
MC SP0531
MC SP0532
MC SP0533
MC SP0534
MC SP0535
MC SP0536
MC SP0537
MC SP0538
MC SP0539
MC SP0540
MC SP0541
MC SP0542
MC SP0543
MC SP0544
MC SP0545
MC SP0546
MC SP0547
MC SP0548
MC SP0549
MC SP0550
MC SP0551
MC SP0552

```

C 71 IF (IMTPLT.NE.1) GO TO 74
C      DO 73 I=1,N
C
C      DO 72 K=1,NSAM
C      72 YP(K) = XM(I,K)
C
C      WRITE (6,156)
C      CALL PLOTP (XP,YP,NSAM,0)
C      73 WRITE (6,161) I
C
C      74 IF (ISMPLT.NE.1) GO TO 77
C
C      DO 76 I=1,N
C
C      DO 75 K=1,NSAM
C      75 YP(K) = ERR(I,K)
C
C      WRITE (6,156)
C      CALL PLOTP (XP,YP,NSAM,0)
C      76 WRITE (6,162) I,I
C
C      77 IF (ISVPLT.NE.1) GO TO 80
C
C      DO 79 I=1,N
C
C      DO 78 K=1,NSAM
C      78 YP(K) = VAR(I,I,K)
C
C      WRITE (6,156)
C      CALL PLOTP (XP,YP,NSAM,0)
C      79 WRITE (6,163) I
C
C      80 WRITE (6,156)
C      CONTINUE
C *****
C      THE FOLLOWING SECTION IS TESTING ONLY
C
C      ONCE TESTING IS COMPLETED.
C
C      500 FORMAT (I1)
C
C      READ (5,500) IT
C      IF (IT.NE.1) GO TO 501
C      DO 517 K=1,NSAM
C      517 READ(5,520) (XS(I,K),I=1,N)
C      520 FORMAT (4(1PE20.12))

```

```

C      DO 502 K=1,NSAM
C      DO 502 IK=1,N
502    READ (5,520) (PKKS(IK,J,K),J=1,N)
C
C      WRITE (6,504)
504    FORMAT (2X,'THE MEANS OF EST. ERROR CALC. FROM EVAL ARE',/)
C
C      DO 518 K=1,NSAM
518    WRITE (6,505) K, (XS(I,K),I=1,N)
505    FORMAT (2X,'K=',I3,5X,6(1PE12.5),/)
C
C      DO 503 K=1,NSAM
C      DO 503 IK=1,N
503    WRITE (6,514) (PKKS(IK,J,K),J=1,N)
C
C      COMPUTE RMS ERRORS OF STATISTICS
C
C      DO 509 I=1,N
C      X(I)=0.
C      DO 509 J=1,N
509    GK(I,J)=0.
C
C      DO 510 I=1,N
C      DO 510 J=1,N
C      DO 510 K=1,NSAM
510    GK(I,J)=GK(I,J)+VAR(I,J,K)**2+PKKS(I,J,K)**2-2.*VAR(I,J,K)*PKKS
      .(I,J,K)
C
C      DO 521 I=1,N
C      DO 521 K=1,NSAM
521    X(I)=X(I)+ERR(I,K)**2+XS(I,K)**2-2.*ERR(I,K)*XS(I,K)
C
C      ENSAM=NSAM
C      DO 511 I=1,N
C      X(I)=SQRT(X(I)/ENSAM)
C      DO 511 J=1,I
511    GK(I,J)=SQRT(GK(I,J)/ENSAM)
C
C      WRITE (6,512) (X(I),I=1,N)
C      WRITE (6,515)
515    FORMAT (5X,'THE RMS DEVIATIONS OF EST ERROR VARIANCES ARE',/)
C
C      DO 513 I=1,N
513    WRITE (6,514) (GK(I,J),J=1,I)
512    FORMAT (5X,'RMS DEVIATIONS OF EST ERROR MEANS ARE',/,9(2X,1PE12.5)
      ,/)
514    FORMAT (9(2X,1PE12.5),/)

```


501 CONTINUE
STOP

C

```
81 FORMAT (5(110))
82 FCFORMAT (12)
83 FCFORMAT (7(110))
84 FCFORMAT (2(15))
85 FCFORMAT (5(110))
86 FCFORMAT (915)
87 FCFORMAT (20X, 'DESCRIPTION OF RUN',//)
88 FCFORMAT (10X, 'GAINS COMPUTED OFF-LINE AND READ IN',//)
89 FCFORMAT (10X, 'GAINS COMPUTED ONCE IN "GAIN" BEFORE STARTING MONTE CARLO',//)
90 FCFORMAT (10X, 'GAINS COMPUTED FOR EACH MEMBER OF ENSEMBLE',//)
91 FCFORMAT (10X, 'THE STANDARD LINEAR EQS. DO NOT CHARACTERIZE THE FILTER',//)
92 FCFORMAT (10X, 'THE STD. KALMAN EQS. CHARACTERIZE THE LINEAR FILTER',//)
93 FCFORMAT (10X, 'ONLY ONE TRACK IS USED AND IT IS GENERATED BY SUBROUTINE TRACK',//)
94 FCFORMAT (10X, 'ONLY ONE TRACK IS USED AND IT IS READ IN',//)
95 FCFORMAT (10X, 'SEVERAL TRACKS USED BUT NOT GENERATED FROM STD. LINEAR DIFFERENCE EQS.',//)
96 FCFORMAT (10X, 'SEVERAL TRACKS GENERATED BY USING THE STD. LINEAR DIFFERENCE EQS.',//)
97 FCFORMAT (10X, 'MEAN OF TRACK, MEAN OF EST. ERROR AND COVARIANCE OF EST. ERROR ARE COMPUTED',//)
98 FCFORMAT (10X, 'MEAN OF TRACK, MEAN AND VARIANCES OF EST. ERROR ARE COMPUTED',//)
99 FCFORMAT (10X, 'THE Q MATRIX IS COMPUTED ON-LINE AT EACH SAMPLE BY "QON"',//)
100 FCFORMAT (10X, 'THE COVARIANCE OF W IS READ IN AND Q IS COMPUTED BY "QMAT" BEFORE STARTING MONTE CARLO',//)
101 FCFORMAT (10X, 'THE Q MATRIX IS READ IN',//)
102 FCFORMAT (10X, 'R IS COMPUTED ON-LINE AT EACH SAMPLE BY "RON"',//)
103 FCFORMAT (10X, 'R IS READ IN',//)
104 FCFORMAT (10X, '20X, 'INPUT DATA CALLED FOR',//)
105 FCFORMAT (10X, 'PHI MATRIX',//)
106 FCFORMAT (10X, 'H MATRIX',//)
107 FCFORMAT (10X, 'R MATRIX',//)
108 FCFORMAT (10X, 'COVARIANCE OF W',//)
109 FCFORMAT (10X, 'Q MATRIX',//)
110 FCFORMAT (10X, 'GAMMA MATRIX',//)
111 FCFORMAT (10X, 'STANDARD DEVIATIONS OF MEASUREMENT NOISE',//)
112 FCFORMAT (10X, 'STANDARD DEVIATIONS OF INPUT FORCING W',//)
113 FCFORMAT (10X, 'XHAT(0/-1)',//)
114 FCFORMAT (10X, 'MEANS AND VARIANCES OF X(0)',//)
115 FCFORMAT (10X, 'P(0/-1)',//)
```

MCSP05553
MCSP05554
MCSP05555
MCSP05556
MCSP05557
MCSP05558
MCSP05559
MCSP05560
MCSP05561
MCSP05562
MCSP05563
MCSP05564
MCSP05565
MCSP05566
MCSP05567
MCSP05568
MCSP05569
MCSP05570
MCSP05571
MCSP05572
MCSP05573
MCSP05574
MCSP05575
MCSP05576
MCSP05577
MCSP05578
MCSP05579
MCSP05580
MCSP05581
MCSP05582
MCSP05583
MCSP05584
MCSP05585
MCSP05586
MCSP05587
MCSP05588
MCSP05589
MCSP05590
MCSP05591
MCSP05592
MCSP05593
MCSP05594
MCSP05595
MCSP05596
MCSP05597
MCSP05598
MCSP05599

MCSP0672
MCSP0673
MCSP0674
MCSP0675
MCSP0676
MCSP0677
MCSP0678
MCSP0679
MCSP0680
MCSP0681
MCSP0682
MCSP0683
MCSP0684
MCSP0685
MCSP0686
MCSP0687
MCSP0688
MCSP0689
MCSP0690

UU U U UU UU U


```

C      CALL VSUB(Z,C,M,VTMP)
C      CALL VPROD(GK,VTMP,N,M,VTMP)
C      CALL VADD(XHKKM1,VTMP,N,XHKK)

C      CALL XPRED
C      RETURN
3      CONTINUE
C      IF STANDARD EQUATIONS ARE NOT TO BE USED, THE APPROPRIATE
C      EQUATIONS MUST BE INSERTED HERE BY THE USER.
C      RETURN
C      END
C      SUBROUTINE GAIN
C      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
C      COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),
1      TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),
2      VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),
3      GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),
4      SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VTMP(5),Z(5),V(5),SIGV(5),
5      XHATZ(5),C(6),IEXT,IPHI,IH,
6      NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M

C      G(K) = P(K/K-1)*HT*(H*P(K/K-1)*HT + R)
C      IF(IH.NE.O) CALL HKK
C      CALL TRANS (H,M,N,TEMP2)
C      CALL PROD (PKKM1,TEMP2,N,N,M,TEMP)
C      CALL PROD (H,TEMP,M,N,M,TEMP1)
C      CALL ADD (TEMP1,R,M,M,TEMP1)
C      IF (M.EQ.1) GO TO 2
C      MD = ND
C      CALL GAUSS3 (M,EPS,TEMP1,TEMP2,KER,MD)
C      CALL PROD (TEMP,TEMP2,N,M,M,G)

C      NOTE HERE   PKK(I,J) = P(K/K)   WHERE
C      P(K/K) = (I-G(K)*H)*P(K/K-1)
1      IF (IEXT.EQ.O) GO TO 4
C      CALL ESTIM
C      IF(IPHI.NE.O) CALL AK
4      CALL PROD (G,H,N,M,N,TEMP)
C      CALL SUB (EI,TEMP,N,N,TEMP2)
C      CALL PROD (TEMP2,PKKM1,N,N,N,PKK)

C      NOTE HERE   PKKMI(I,J) = P(K/K-1)   WHERE
C      P(K/K-1) = PHI*P(K-1/K-1)*PHIT + Q
C      IF(IQ.EQ.-1) CALL QON
C      CALL TRANS (PHI,N,N,TEMP2)
C      CALL PROD (PKK,TEMP2,N,N,N,TEMP)

```

MCSP0691
MCSP0692
MCSP0693
MCSP0694
MCSP0695
MCSP0696

MCSP0704
MCSP0705
MCSP0706
MCSP0707
MCSP0708
MCSP0709
MCSP0710
MCSP0711
MCSP0712
MCSP0713
MCSP0714
MCSP0715
MCSP0716

MCSP0717
MCSP0718
MCSP0719
MCSP0720
MCSP0721
MCSP0722
MCSP0723
MCSP0724

CALL PROD (PHI,TEMP,N,N,N,TEMP1)	MCSP0725
CALL ADD (TEMP1,Q,N,N,PKKM1)	MCSP0726
RETURN	MCSP0727
DO 3 I=1,N	MCSP0728
3 G(I,1) = TEMP(I,1)/TEMP1(1,1)	MCSP0729
GO TO 1	MCSP0730
END	MCSP0731
SUBROUTINE QMAT	MCSP0732
THIS SUBROUTINE COMPUTES THE MATRIX Q FROM THE EQUATION	MCSP0733
Q=GAMMA* E(W*WT) * GAMMAT	MCSP0764
DOUBLE PRECISION ARITHMETIC IS USED	MCSP0765
REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI	MCSP0766
COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),	MCSP0767
1TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),	MCSP0768
2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),	MCSP0769
3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),	MCSP0770
4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VTMP(5),Z(5),V(5),SIGV(5),	MCSP0771
5XHATZ(5),C(6),IEXT,IPHI,IH,	MCSP0772
6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M	MCSP0773
CALL PROD (GAMMA,COVW,N,IN,IN,TEMP)	MCSP0781
CALL TRANS (GAMMA,N,IN,TEMP)	MCSP0782
CALL PROD (TEMP,TEMP1,N,IN,N,Q)	MCSP0783
RETURN	MCSP0784
END	MCSP0785
SUBROUTINE QQN	MCSP0786
IF Q IS TO BE COMPUTED ON-LINE (IFLQ.NE.0) IT IS DONE	MCSP0787
IN THIS SUBROUTINE	MCSP0788
REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI	MCSP0789
COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),	MCSP0790
1TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),	MCSP0791
2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),	MCSP0792
3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),	MCSP0793
4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VTMP(5),Z(5),V(5),SIGV(5),	
5XHATZ(5),C(6),IEXT,IPHI,IH,	
6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M	
THE APPROPRIATE STATEMENTS FOR COMPUTING Q ON-LINE MUST	MCSP0801
	MCSP0802

MCSP0803
MCSP0804

```
C BE INSERTED HERE BY THE USER
C
C REAL*8 DELT,DELT1,SIGVS,SIGTS,SIGFS,A,B(5)
C
C DO 1 I=1,N
C 1 B(I)=XHKK(I)
C
C DELT=100.D0
C SIGVS=31.7D-06
C SIGTS=3.04D-06
C SIGFS=.25D-06
C A=-1.D0
C DELT1=DELT**2
C
C Q(1,1)=(B(2)**2)*DELT1*SIGTS
C Q(1,2)=-((B(1)*B(2)*DELT1*SIGTS)
C Q(1,3)=0.D0
C Q(1,4)=A*B(2)*DELT1*SIGTS
C Q(1,5)=0.D0
C Q(2,2)=((DELT1**2)/4.D0)*SIGVS+(B(1)**2)*DELT1*SIGTS
C Q(2,3)=((DELT1**3)/2.D0)*SIGVS
C Q(2,4)=-((A*B(1)*DELT1*SIGTS)
C Q(2,5)=0.D0
C Q(3,3)=DELT1*SIGVS
C Q(3,4)=0.D0
C Q(3,5)=0.D0
C Q(4,4)=DELT1*SIGTS
C Q(4,5)=0.D0
C Q(5,5)=DELT1*SIGFS
C Q(3,1)=Q(1,2)
C Q(3,2)=Q(2,3)
C Q(4,1)=Q(1,4)
C Q(4,2)=Q(2,4)
C Q(4,3)=Q(3,4)
C Q(5,1)=Q(1,5)
C Q(5,2)=Q(2,5)
C Q(5,3)=Q(3,5)
C Q(5,4)=Q(4,5)
C
C RETURN
C END
C SUBROUTINE RON
C
C IF R IS TO BE COMPUTED ON-LINE (IFLR.NE.0) IT IS DONE
C IN THIS SUBROUTINE
C
C REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
```

MCSP0805
MCSP0806
MCSP0807
MCSP0808
MCSP0809
MCSP0810
MCSP0811
MCSP0812


```

COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),
1TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),
2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),
3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),
4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VIMP(5),Z(5),V(5),SIGV(5),
5XHATZ(5),C(6),IEXT,IPHI,IH,
6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M

THE APPROPRIATE STATEMENTS FOR COMPUTING R ON-LINE MUST
BE INSERTED HERE BY THE USER

RETURN
END
SUBROUTINE STAT
THIS SUBROUTINE COMPUTES RUNNING SUMS USED IN DETERMINING THE
SAMPLE STATISTICS OF TRACK AND ESTIMATION ERRORS. IN THE DEFAULT
OPTION (ISTAT.EQ.0) THE STATISTICS TO BE COMPUTED ARE MEAN OF
TRACK, MEAN OF ESTIMATION ERROR AND VARIANCE OF ESTIMATION
ERROR. IF (ISTAT.NE.0) THE OFF-DIAGONAL TERMS IN THE COVARIANCE OF
ESTIMATION ERROR MATRIX ARE ALSO COMPUTED.
REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),
1TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),
2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),
3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),
4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VIMP(5),Z(5),V(5),SIGV(5),
5XHATZ(5),C(6),IEXT,IPHI,IH,
6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M
DIMENSION EXH(5)
IF (ITRK.NE.1) GO TO 2
IF (ITER.NE.1) GO TO 4
DO 1 J=1,N
1 XM(J,K) = XS(J,K)
GO TO 4
2 CONTINUE
DO 3 J=1,N
3 XM(J,K) = XM(J,K)+XS(J,K)
4 CONTINUE
DO 5 J=1,N
EXH(J) = XHKK(J)-XS(J,K)
ERR(J,K) = ERR(J,K)+EXH(J)
5 VAR(J,J,K) = VAR(J,J,K)+EXH(J)**2

```

MCSP0820
MCSP0821
MCSP0822
MCSP0823
MCSP0824
MCSP0825
MCSP0826
MCSP0827
MCSP0828
MCSP0829
MCSP0830
MCSP0831
MCSP0832
MCSP0833

MCSP0842
MCSP0843
MCSP0844
MCSP0845
MCSP0846
MCSP0847
MCSP0848
MCSP0849
MCSP0850
MCSP0851
MCSP0852
MCSP0853
MCSP0854
MCSP0855
MCSP0856
MCSP0857
MCSP0858
MCSP0859
MCSP0860


```

C      IF (ISTAT.EQ.0) RETURN
C      DO 6 L=2,N
C      LM1 = L-1
C
C      DO 6 J=1,LM1
C      6 VAR(L,J,K) = VAR(L,J,K)+EXH(L)*EXH(J)
C
C      RETURN
C      END
C      SUBROUTINE XZERO
C      THIS SUBROUTINE GENERATES THE INITIAL STATE VALUE FROM A NORMAL
C      RANDOM NUMBER GENERATOR. IT IS ASSUMED THAT THE INITIAL STATE
C      HAS COMPONENTS THAT ARE INDEPENDENT
C      REAL*8 GAMMA,CQVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
C      COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),CQVW(5,5),
C      1 TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),
C      2 VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),
C      3 GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),X(5),
C      4 SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VTMP(5),Z(5),V(5),SIGV(5),
C      5 XHATZ(5),C(6),IEXT,I PHI,IH,
C      6 N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M
C      CALL SNORM (IXZ,X,N)
C
C      DO 1 I=1,N
C      1 XS(I,1) = SIGXZ(I)*X(I)+XZMEAN(I)
C
C      RETURN
C      END
C      SUBROUTINE ADD (A,B,N,M,C)
C      THIS SUBROUTINE ADDS THE NXM MATRICES A AND B, STORING THE
C      RESULT IN C
C      REAL*8 A,B,C
C      DIMENSION A(5,5),B(5,5),C(5,5)
C
C      DO 1 I=1,N
C      DO 1 J=1,M
C      1 C(I,J) = A(I,J)+B(I,J)
C
C      RETURN
C      END
C      SUBROUTINE MREAD (A,N,M)
C      THIS SUBROUTINE READS AN NXM MATRIX A ACCORDING TO THE FORMAT
C      8D10.5. THE ENTRIES IN THE FIRST ROW OF A ARE READ FIRST, THEN
C      THE ENTRIES IN THE SECOND ROW, AND SO ON.
C      REAL*8 A
C      DIMENSION A(5,5)

```

MCSP0861
 MCSP0862
 MCSP0863
 MCSP0864
 MCSP0865
 MCSP0866
 MCSP0867
 MCSP0868
 MCSP0869
 MCSP0870
 MCSP0934
 MCSP0935
 MCSP0936
 MCSP0937
 MCSP0938

MCSP0946
 MCSP0947
 MCSP0948
 MCSP0949
 MCSP0950
 MCSP0951
 MCSP0952
 MCSP0953
 MCSP0954
 MCSP0955
 MCSP0956
 MCSP0958
 MCSP0959
 MCSP0960
 MCSP0961
 MCSP0962
 MCSP0963
 MCSP0964
 MCSP0965
 MCSP0966
 MCSP0967
 MCSP0968
 MCSP0969
 MCSP0970


```

C      DO 1 I=1,N
C      1 READ (5,2) (A(I,J),J=1,M)
C      RETURN
C      2 FORMAT (8F10.0)
C      END
C      SUBROUTINE MWRITE (A,N,M)
C      THIS SUBROUTINE WRITES THE ENTRIES OF THE NXM MATRIX A
C      REAL*8 A
C      DIMENSION A(5,5)
C
C      DO 1 I=1,N
C      1 WRITE (6,2) (A(I,J),J=1,M)
C      RETURN
C      2 FORMAT (9(2X,1PE12.5))
C      END
C      SUBROUTINE PROD (A,B,N,M,L,C)
C      THIS SUBROUTINE COMPUTES THE MATRIX PRODUCT AB AND STORES THE
C      RESULT IN C
C      A = NXM, B = MXL, C = NXL
C      REAL*8 A,B,C,T
C      DIMENSION A(5,5),B(5,5),C(5,5),T(5,5)
C
C      DO 1 I=1,N
C      DO 1 J=1,L
C      1 T(I,J) = 0.0
C
C      DO 2 I=1,N
C      DO 2 J=1,L
C      DO 2 K=1,M
C      2 T(I,J) = T(I,J)+A(I,K)*B(K,J)
C
C      DO 3 I=1,N
C      DO 3 J=1,L
C      3 C(I,J) = T(I,J)
C      RETURN
C      END

```

MCSP0972
 MCSP0973
 MCSP0974
 MCSP0975
 MCSP0976
 MCSP0977
 MCSP0978
 MCSP0979
 MCSP0980
 MCSP0981
 MCSP0982

MCSP0984
 MCSP0985
 MCSP0986
 MCSP0987
 MCSP0988
 MCSP0989
 MCSP0990
 MCSP0991
 MCSP0992
 MCSP0993
 MCSP0994
 MCSP0995
 MCSP0996

MCSP0998
 MCSP0999
 MCSP1000
 MCSP1001
 MCSP1002
 MCSP1003
 MCSP1004
 MCSP1005
 MCSP1006
 MCSP1007
 MCSP1008
 MCSP1009
 MCSP1010
 MCSP1011
 MCSP1012
 MCSP1013
 MCSP1014
 MCSP1015
 MCSP1016
 MCSP1017
 MCSP1018
 MCSP1019

C	C	SUBROUTINE SUB (A,B,N,M,C)	MCSP1020
C	C	THIS SUBROUTINE SUBTRACTS THE NXM MATRIX B FROM THE NXM MATRIX	MCSP1021
		A AND STORES THE RESULT IN C	MCSP1022
		REAL*8 A,B,C	MCSP1023
C		DIMENSION A(5,5),B(5,5),C(5,5)	MCSP1025
C		DO 1 I=1,N	MCSP1026
C		DO 1 J=1,M	MCSP1027
	1	C(I,J) = A(I,J)-B(I,J)	MCSP1028
C		RETURN	MCSP1029
		END	MCSP1030
		SUBROUTINE TRANS (A,N,M,C)	MCSP1031
C	C	THIS SUBROUTINE FORMS THE MATRIX TRANSPOSE OF A STORING THE	MCSP1032
C	C	RESULT IN C	MCSP1033
		A = NXM, C = MXN	MCSP1034
		REAL*8 A,C	MCSP1035
		DIMENSION A(5,5),C(5,5)	MCSP1036
C		DO 1 I=1,N	MCSP1037
C		DO 1 J=1,M	MCSP1039
	1	C(J,I) = A(I,J)	MCSP1040
C		RETURN	MCSP1041
		END	MCSP1042
		SUBROUTINE VADD (X,Y,N,Z)	MCSP1043
C	C	THIS SUBROUTINE COMPUTES THE SUM OF THE N-VECTORS X AND	MCSP1044
C	C	Y AND STORES THE RESULT IN THE N-VECTOR Z	MCSP1045
		REAL*4 X(5),Y(5),Z(5)	MCSP1046
C		DO 1 I=1,N	MCSP1047
	1	Z(I) = X(I)+Y(I)	MCSP1048
C		RETURN	MCSP1049
		END	MCSP1050
		SUBROUTINE VPROD (A,X,M,N,Y)	MCSP1052
C	C	THIS SUBROUTINE COMPUTES THE PRODUCT OF THE MXN MATRIX	MCSP1053
		A AND THE N-VECTOR X AND STORES THE RESULT IN THE	MCSP1054
		M-VECTOR Y	MCSP1055
		REAL*4 A(5,5),X(5),Y(5),T(5)	MCSP1057
C		DO 1 I=1,M	MCSP1058
C		T(I) = 0.00	MCSP1059
C			MCSP1060
C			MCSP1061
C			MCSP1062
C			MCSP1063
			MCSP1065
			MCSP1066
			MCSP1067


```

C      DO 1 J=1,N
C      1 T(I) = T(I)+A(I,J)*X(J)
C
C      DO 2 I=1,M
C      2 Y(I) = T(I)
C      RETURN
C      END
C      SUBROUTINE VREAD (V,N)
C      THIS SUBROUTINE READS THE N-DIMENSIONAL S.P. VECTOR V
C
C      DIMENSION V(5)
C      READ (5,1) (V(I),I=1,N)
C      RETURN
C
C      1 FORMAT (8F10.0)
C      END
C      SUBROUTINE VSUB (X,Y,N,Z)
C      THIS SUBROUTINE COMPUTES THE DIFFERENCE X-Y OF THE TWO
C      N-VECTORS X & Y AND STORES THE RESULT IN THE N-VECTOR Z
C
C      REAL*4 X(5),Y(5),Z(5)
C
C      DO 1 I=1,N
C      1 Z(I) = X(I)-Y(I)
C      RETURN
C      END
C      SUBROUTINE VWRITE (V,N)
C      THIS SUBROUTINE WRITES THE N-DIMENSIONAL S.P. VECTOR V
C
C      DIMENSION V(5)
C      WRITE (6,1) (V(I),I=1,N)
C      RETURN
C
C      1 FORMAT (9(2X,1PE12.5))
C      END
C      SUBROUTINE MEAS
C      THIS SUBROUTINE MEAS STARTS WITH THE TRUE STATE VALUE XS
C      AND ADDS ZERO-MEAN WHITE GAUSSIAN NOISE TO H*XS TO
C      GENERATE A NOISY VECRCTOR OF MEASUREMENTS Z.
C
C      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI

```

```

MCSP1068
MCSP1069
MCSP1070
MCSP1071
MCSP1072
MCSP1073
MCSP1074
MCSP1075
MCSP1076
MCSP1077
MCSP1078
MCSP1079
MCSP1080
MCSP1081
MCSP1083
MCSP1084
MCSP1085
MCSP1086
MCSP1087
MCSP1088
MCSP1089
MCSP1090
MCSP1091
MCSP1093
MCSP1094
MCSP1095
MCSP1096
MCSP1097
MCSP1098
MCSP1099
MCSP1100
MCSP1101
MCSP1102
MCSP1104
MCSP1105
MCSP1106
MCSP1107
MCSP1108
***00910
***00920
***00930
***00940
***00950
***00960

```



```

COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),
1TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),
2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),
3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),
4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VTMP(5),Z(5),V(5),SIGV(5),
5XHATZ(5),C(6),IEXT,IPHI,IH,
6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M

VP=1640.
A=-1.
P=3.1416
CALL SNORM(IV,V,M)

V(1)=SIGV(1)*V(1)
V(2)=SIGV(2)*V(2)

Z(1)=XS(4,K)-ATAN(A*XS(1,K)/XS(2,K))

Z(1)=Z(1)+V(1)
IF(XS(2,K).LT.0.) Z(1)=Z(1)-P

R1=(XS(1,K)**2)+(XS(2,K)**2)
R1=SQRT(R1)

Z(2)=VP+(XS(3,K)*XS(2,K))/R1
Z(2)=(XS(5,K)*VP)/Z(2)
Z(2)=Z(2)+V(2)
RETURN
END
SUBROUTINE TRACK
IF TRACK IS TO BE GENERATED ON-LINE IT IS DONE IN THIS SUBROUTINE
IN THE DEFAULT OPTION (ITRK.EQ.0) THE TRACK IS GENERATED
FROM THE STANDARD LINEAR DIFFERENCE EQUATION

X(K+1)=PHI*X(K)+GAMMA*W(K)

REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),
1TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),
2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),
3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),
4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VTMP(5),Z(5),V(5),SIGV(5),
5XHATZ(5),C(6),IEXT,IPHI,IH,
6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M
DIMENSION W(5)
ITRK NE.0 OR 1 -- SEVERAL TRACKS GENERATED,BUT NOT FROM STD.
LINEAR EQS.
= 0 -- SEVERAL TRACKS GENERATED FROM STD LINEAR EQS

***02360
***02370
***02380
***02390
***02400
***02410
***02420
***02430

***02520
***02530
***02540

```



```

C      = 1 -- ONLY ONE TRACK IS USED
C      IF(ITRK.NE.O) GO TO 100
C      CALL SNORM(IW,W,IN)
C      CONVERT EACH V(O,1) R.V. TO N(O,SIGW(I)) R.V.
DO 1 I=1,IN
1  W(I)=SIGW(I)*W(I)
DO 3 I=1,N
3  X(I)=XS(I,K)
C      CALL VPROD(GAMMAS,W,N,IN,W)
C      CALL VPROD(PHIS,X,N,N,VTMP)
C      CALL VADD(VTMP,W,N,VTMP)
DO 2 I=1,N
2  XS(I,K+1)=VTMP(I)
C      NEW VALUE OF X HAS BEEN COMPUTED AND STORED IN THE ARRAY XS
RETURN
100 IF(ITRK.NE.1) GO TO 200
C      IF(ITRK.EQ.1) THE USER MUST INSERT HERE THE STATEMENTS REQUIRED
C      TO GENERATE A SINGLE TRAJECTORY AND STORE IT IN THE ARRAY XS
C      XS(I,K), I=1,N,K=2,NSAM (NOTE THAT IF A SINGLE TRAJECTORY IS TO BE
C      GENERATED, THE INITIAL CONDITION HAS BEEN READ IN AND STORED
C      IN XS(I,1), I=1,N)
C      DC 5 K=2,NSAM
C      DO 4 I=1,N
C      4  X(I)=XS(I,K-1)
C      CALL VPROD(PHIS,X,N,N,X)
C      DO 5 I=1,N
C      XS(I,K)=X(I)
C      5  CONTINUE
C      RETURN
C      CONTINUE
C      IF THIS POINT IS REACHED, ITRK NOT EQUAL 0 OR 1 INDICATING THAT
C      SEVERAL TRACKS ARE TO BE GENERATED, BUT NOT BY USING THE STD.
C      LINEAR DIFFERENCE EQS..THE USER MUST SUPPLY THE APPROPRIATE
C      STATEMENTS HERE.
C      RETURN
C      END
C      SUBROUTINE XPRED
C      REAL*8 GAMMA,COVM,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
C      COMMON EI(5,5),Q(5,5),PKK(5,5),GAMMA(5,5),COVM(5,5),
C      1TEMP(5,5),TEMP1(5,5),H(5,5),PKKM1(5,5),CR(5,5),PHI(5,5),
C      2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),
C      3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),
C      4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VTMP(5),Z(5),V(5),SIGV(5),
C      5XHATZ(5),C(6),IEXT,IPHI,IH,

```

```

***02550
***02560
***02580
***02590
***02600
***02610
***02620
***02630
***02640
***02650
***02660
***02670
***02680
***02740
***02750
***02760
***02770
***02780
***02790
***02800

```

```

***02810
***02820
***02830
***02840
***02850
***02860
***02870
***02880
MCSP0740

```



```
6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M
```

```
THIS SUBROUTINE CALCULATES THE STATE PREDICTION  
MATRIX XHAT(K+1/K)=A(XHAT(K/K),U(K),K)  
AND STORES IN THE ARRAY XHKKM1(I)
```

```
CALL VPROD(PHIS,XHKK,N,N,XHKKM1)  
RETURN  
END
```

```
SUBROUTINE CK  
REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI  
COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),  
1TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),  
2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),  
3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),  
4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VTMP(5),Z(5),V(5),SIGV(5),  
5XHATZ(5),C(6),IEXT,IPHI,IH,  
6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M
```

MCSP0878

```
THIS SUBROUTINE CALCULATES THE CURRENT VALUE OF MATRIX C(X(K/K-1))
```

```
A=-1.  
P=3.1416  
VP=1640.  
IF(Z(2).LT.500.) GO TO 1  
IF(XHKKM1(2).GT.0.) XHKKM1(2)=-XHKKM1(2)  
GO TO 2  
1 IF(XHKKM1(2).LT.0.) XHKKM1(2)=-XHKKM1(2)  
2 C(1)=XHKKM1(4)-ATAN(A*XHKKM1(1)/XHKKM1(2))  
IF(XHKKM1(2).LT.0.) C(1)=C(1)-P  
R1=(XHKKM1(1)**2)/(XHKKM1(2)**2)  
R1=SQRT(R1)  
C(2)=VP+(XHKKM1(3)*XHKKM1(2))/R1  
C(2)=(VP*XHKKM1(5))/C(2)  
RETURN  
END
```

```
SUBROUTINE AK
```

```
REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI  
COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),  
1TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),  
2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),  
3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),  
4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VTMP(5),Z(5),V(5),SIGV(5),  
5XHATZ(5),C(6),IEXT,IPHI,IH,  
6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M
```



```

C THIS SUBROUTINE CALCULATES THE A(K)=PHI(I,J) BY USING XHAT(K/K)
C FOR EXTENDED KALMAN FILTER
C
C REAL*8 A(4)
C
C DO 1 I=1,N
C 1 A(I)=XHKK(I)
C RETURN
C END
C SUBROUTINE HKK
C
C THIS SUBROUTINE CALCULATES THE CURRENT LINEARIZED MEASUREMENT MATRIX
C H(K) BY USING X(K/K-1)=XHKKM1(I)
C REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI
C COMMON EI(5,5),Q(5,5),G(5,5),PKK(5,5),GAMMA(5,5),COVW(5,5),
C 1TEMP(5,5),TEMP1(5,5),TEMP2(5,5),H(5,5),PKKM1(5,5),R(5,5),PHI(5,5),
C 2VAR(5,5,60),GKS(5,5,60),PKKS(5,5,60),XM(5,60),ERR(5,60),
C 3GAMMAS(5,5),PHIS(5,5),XS(5,60),HS(5,5),GK(5,5),SIGW(5),X(5),
C 4SIGXZ(5),XZMEAN(5),XHKK(5),XHKKM1(5),VIMP(5),Z(5),V(5),SIGV(5),
C 5XHATZ(5),C(6),IEXT,IPHI,IH,
C 6N,NSAM,IQ,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,M
C REAL*8 A,VP,B(5),R1,F
C A=-1.00
C VP=1640.00
C DO 1 I=1,N
C 1 B(I)=XHKKM1(I)
C
C R1=(B(1)**2)+(B(2)**2)
C R1=DSQRT(R1)
C
C F=VP+(B(3)*B(2))/R1
C F=(B(5)*VP)/F
C
C H(1,1)=- (A*B(2))/(R1**2)
C H(1,2)=(A*B(1))/(R1**2)
C H(1,3)=0.00
C H(1,4)=1.00
C H(1,5)=0.00
C
C H(2,1)=(F**2)*B(3)*B(1)*B(2)
C H(2,1)=H(2,1)/(B(5)*VP*(R1**3))
C H(2,2)=-((F**2)*B(3)*(B(1)**2))
C H(2,2)=H(2,2)/((R1**3)*VP*B(5))
C H(2,3)=-((F**2)*B(2))
C H(2,3)=H(2,3)/(B(5)*VP*R1)

```


C
H(2,4)=0.D0
H(2,5)=F/B(5)

RETURN
END

BIBLIOGRAPHY

1. D. E. Kirk, State Estimators and Predictors for Fire Control Systems, Naval Postgraduate School report NPS-52K174101, Monterey, California, October 1974.
2. Davenport, Probability and Random Processes, McGraw-Hill, 1970, p. 250-251.
3. Jazwinski, Stochastic Processes and Filtering Theory, Academic Press, 1970, p. 39-41.
4. G. W. Mitschang, An Application of Nonlinear Filtering Theory to Passive Target Location and Tracking, Ph.D. Thesis, Naval Postgraduate School, Monterey, California, 1974.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3. Professor D. E. Kirk, Code 52Ki Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
4. Professor H. A. Titus, Code 52Ts Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
5. Bogazici Universitesi Arnavutkoy, Istanbul, Turkey	1
6. Ortadogu Teknik Universitesi Ankara, Turkey	1
7. Istanbul Teknik Universitesi Istanbul, Turkey	1
8. Toshiaki Iida 153 Terayama Hatano-City Kanagawa, Japan	1
9. LCdr. Marcilio Bpa Cunha Naval Postgraduate School S.M.C. 2379 Monterey, California 93940	1
10. Lt. H. O. Tasdelen Denizalti Filosu, Golecuk Kocaeli, Turkey	2



163226

Thesis
T164
c.1

Thesis
T164
c.1

Tasdelen

Approximate analytical
evaluation of extended-
Kalman filters.

6 MAR 77

5 SEP 78

10 APR 79

10 JUL 82

31 DEC 83

163226

23472

25329

24676

27552

27924

Thesis
T164
c.1

Tasdelen

Approximate analytical
evaluation of extended-
Kalman filters.

163226

thesT164

Approximate analytical evaluation of ext



3 2768 001 01386 5
DUDLEY KNOX LIBRARY